
GIMP » Slackware » Perl » ACL
LXFDVD
Slackware 13
Быстрый, стабильнаый, безопасный: 1000+ пакетов
ПЛЮС: ALT Linux 5.0 » Haiku » Linux-ремикс » Игры
LINUX FORMAT
Главное в мире Linux
Декабрь 2009 № 12 (125)
Linux-ремикс
Собрать свой дистрибутив по силам и гуру, и новичку!
» Linux From Scratch » SUSE Studio » Ubuntu » Arch Linux
36 страниц учебников на любой вкус!
Remix Linux
Directions
1 Preheat your distro to 200c
2 Separate the kernels and mix in GCC.
3 Cover with glibc and sprinkle with Bash.
4 Cook for 20 minutes and add the X server.
Enjoy!!!
Prep 4 mins
Cook 20 mins
Total 24 mins
Ingredients
3 raw kernels
1 tbsp GCC
15 oz glibc
1 pinch Bash
1 X server
124 | Linux Format
4 607130 823897
Домашняя бухгалтерия
Сможет ли KMyMoney покончить с кризисом раз и навсегда? с. 11
Управляйте пакетами
Yum, APT и PackageKit — разберитесь с ними всеми с. 40
С их стороны это вовсе не благотворительность
Джим Землин про Microsoft и Linux с. 28
Плюс!
» Менеджеры коллекций
» Видеомонтаж
» Zend Studio и PHP
» Виртуализация! Познакомьтесь с KVM и его друзьями с. 34
» GIMP! Воссоздайте стиль Города Грехов с. 56
» Python! Google — кладезь знаний. Воспользуйтесь ими! с. 76
Подписной индекс в каталоге агентства «Роспечать» 20882
Подписной индекс в каталоге «Пресса России» 87974
Linux center
www.linuxcenter.ru



Что мы делаем

» Мы поддерживаем открытое сообщество, предоставляя источник информации и площадку для обмена мнениями.

» Мы помогаем всем читателям получить от Linux максимум пользы, публикуя статьи в разделе «Учебники»: здесь каждый найдет что-то по своему вкусу.

» Мы выпускам весь код, появляющийся на страницах раздела «Учебники», по лицензии GNU GPLv3.

» Мы стремимся предоставлять точные, актуальные и непредвзятые сведения обо всем, что касается Linux и свободного ПО.

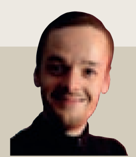
Corporate Patron — Free Software Foundation

Кто мы

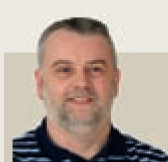
В этом месяце мы создавали свои собственные дистрибутивы Linux, и поэтому спросили у команды **LXF**: а если бы вы переписывали Linux с нуля, что бы вы изменили?



Грэм Моррисон
Лицензию – все понимают, что Beerware лучше, чем Free Software. Ммм, пиво...



Майк Сондерс
Он бы работал в реальном 16-битном режиме, как лучшая открытая операционка – MikeOS!



Нейл Ботвик
Я бы потратил миллион на приветственную мелодию. Бог с ним, с кодом. Ба-да-да-дам!



Эфраин Эрнандес-Мендоса
Мне, как дизайнеру, необходим индикатор загрузки, охватывающий все 32 моих монитора.



Эндрю Грегори
Поддержку джойстика в *Grub*. Нажмите три кнопки – и вы в секретной комнате (runlevel 99).



Энди Ченнел
Он бы загружался, как старые игры ZX Spectrum: с цветными полосками и пиликаньем.



Дэвид Картрайт
Дал бы имя, которое он заслуживает: ура GNU/BSD/Perl/X11/KDE/Gnome/Bash/ls/что-то-там/Linux!



Энди Хадсон
Я бы ломал стереотипы, заставляя текст прокручиваться вниз, а не вверх. Бойтесь меня!



Ник Вейч
Портировал бы его на действительно стоящее оборудование. Ну, скажем, машину Бэббиджа.



Сюзан Линтон
Вставила бы видео Стива Балмера, танцующего манки, при старте системы. Да, детка, зажигай!



Шашанк Шарма
Во-первых, переименовал бы его: Linux Turbo Championship Edition Pro 2050 – это круто.



Маянк Шарма
Больше пасхальных яиц! Даешь полноценный имитатор полета, встроенный в *sed*!



По вашим правилам

» Не знаю, как вы, а я всегда недолюбливал так называемые «дружественные пользователю» дистрибутивы – еще с тех пор, когда не мог их правильно настроить и не отличал *ls* от *ld*. Не то чтобы они были плохими: я всегда рекомендовал (и продолжаю рекомендовать) их своим менее требовательным знакомым. Однако для меня ситуация всегда развивалась по схеме: поставил – порадовался красоте – начал работать – нашел одну-две-три шероховатости – огорчился – удалил.

В отличие от …, Linux – такая система, которую хочется довести до идеала, а решения, принимаемые за вас разработчиками Ubuntu, Mandriva, Fedora, openSUSE и других систем, призванных работать «из коробки», этому не способствуют, если только вы не среднестатистический пользователь с процессором 1 ГГц, памятью 1 ГБ и винчестером на 1 ТБ. И именно здесь на сцену выходят дистрибутивы-конструкторы, которые можно превратить во что угодно по желанию создателя: от рабочего стола и до встраиваемой системы.

Не скрою, собирать свой дистрибутив сложно, трудоемко, да и не нужно, если в оригинале вас не устраивает отсутствие пары программ и экранные обои (сколько можно наблюдать эти непрерывные ...buntu?). В последнее время мы стали свидетелями рождения нового термина – «респин». *Revisor* от Fedora, SUSE Studio, *Ubuntu Customisation Kit* – эти инструменты позволяют приблизиться к системе вашей мечты, не выходя за рамки родительского дистрибутива. Просто, быстро и удобно.

Хотите узнать, как пользоваться всем этим многообразием? Тогда вперед, на стр. 20! И не забудьте прислать нам ссылочку, если у вас получится что-то стоящее.

Валентин Синицын, главный редактор
» info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Web-сайт: www.linuxformat.ru
› Адрес редакции: Россия, Санкт-Петербург, Лиговский пр., 50, корп. 15
› Телефон редакции: (812) 309-06-86. Дополнительная информация на с. 112

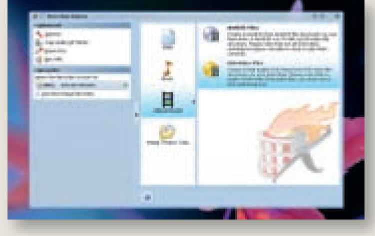


Содержание

Весь номер – прямо как на ладони: приятного чтения!

Обзоры





› Узнайте, что на самом деле думает Грэм о *Nero*, на стр. 10

Сравнение: Менеджеры коллекций

Linux-ремикс

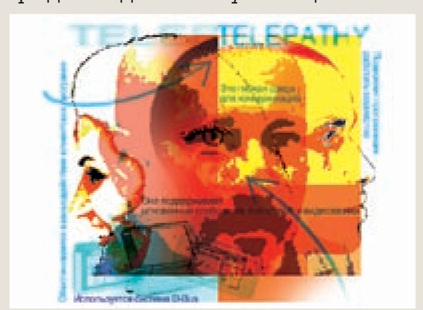


Вы не любите Linux? Да вы просто не умеете его готовить! с. 20

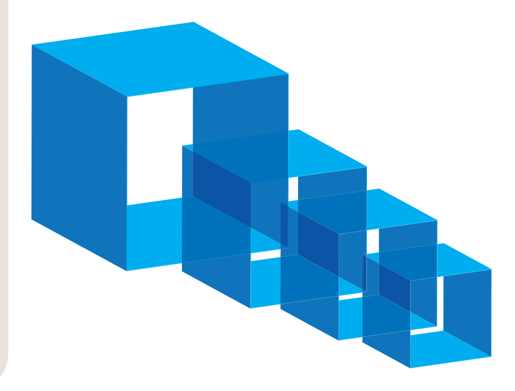
Что за штука...

Telepathy

Мы догадались, что это какое-то средство для коммуникации с. 44



Виртуализация – это просто с. 34



Люди говорят



«Я провожу немало времени в Азии – рост Linux там просто поразителен»

Джим Землин идет на восток с. 28

Slackware 13.0!

LXF DVD125

Простота, стабильность, безопасность

» **Zenwalk 6.2** Шустрый представитель славного семейства
» **Набор для ремикса** Создайте себе свой собственный Linux
» **Haiku OS** Свободный клон BeOS достиг альфа-версии
...плюс ALT Linux 5.0 на второй стороне! с. 100

Ищите в этом номере...

Постоянные рубрики



› **Наша команда всегда рада подписать вас на LXF. Электронная версия в подарок!**

Учебники



ГЛАВНОЕ: ARM все же приходит на рынок нетбуков?
» Google — не новая империя зла? » Революция или эволюция?

Нетбуки на ARM

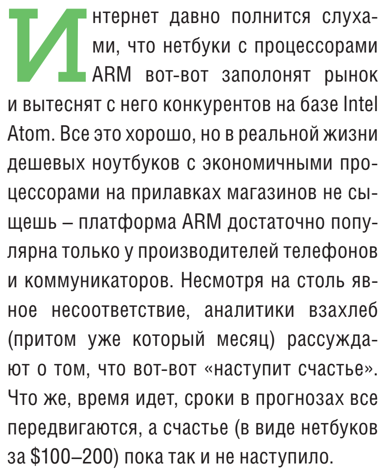
Мифы и реальность: стоит ли ожидать скорого заката эпохи Intel Atom, и действительно ли это поможет Linux продвинуться в сегменте ультрапортативов?

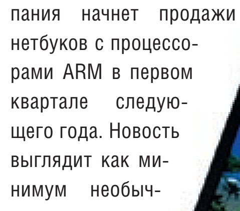


» Рубрику ведет **ЕВГЕНИЙ КРЕСТНИКОВ**

Интернет давно полнится слухами, что нетбуки с процессорами ARM вот-вот заполонят рынок и вытеснят с него конкурентов на базе Intel Atom. Все это хорошо, но в реальной жизни дешевых ноутбуков с экономичными процессорами на прилавках магазинов не сыщешь – платформа ARM достаточно популярна только у производителей телефонов и коммуникаторов. Несмотря на столь явное несоответствие, аналитики взахлеб (притом уже который месяц) рассуждают о том, что вот-вот «наступит счастье». Что же, время идет, сроки в прогнозах все передвигаются, а счастье (в виде нетбуков за $100–200) пока так и не наступило.

Тем не менее, недавно произошло событие, которое заставило нас несколько пересмотреть свою точку зрения на этот вопрос. В Сети появилась информация, что компания ASUSTeK заинтересовалась платформой ARM. Более того, портал www.reghardware.co.uk, со ссылкой на исполнительного директора Asus Джерри Шена [Jerry Shen], сообщил, что компания начнет продажи нетбуков с процессорами ARM в первом квартале следующего года. Новость выглядит как минимум необычно: раньше представители Asus неоднократно высказывали мнение об отсутствии у этой платформы перспектив в сегменте нетбуков. Вообще, отношения с ARM у тайваньского производителя складывались непросто – на Computex 2009 компания представила прототип нетбука, но вскоре тот же Джерри Шен заявил об отсутствии у Asus интереса к продвижению подобных разработок.





> «Едва ли машины на ARM способны потеснить Intel Atom.»

Мы тут же попросили российских представителей компании прокомментировать эту ситуацию, но ответа от них пока нет. Если новость соответствует действительности, это может стать поворотным моментом в развитии нетбуков на ARM – впервые ими заинтересовался столь популярный производитель. Пока подобные машины выпускали малоизвестные фирмы, либо известные компании, для которых производство нетбуков не является профильной областью (например, Sharp PC-Z1 Netwalker). В общем дела шли ни шатко, ни валко, и вдруг – такой сюрприз!

Я бы не стал делать радужных прогнозов – едва ли машины на ARM способны сильно потеснить устройства на Intel Atom. Но, заинтересовав ведущих производителей, они могут неплохо продвинуться в сегменте бюджетных нетбуков. А вместе с ARM в этом сегменте продвинется и Linux: по крайней мере, так принято считать. Хотя, возможно, и корпорация Microsoft не останется в стороне, если новые устройства будут популярны у потребителя. Собственно, запустить Windows CE на ARM можно уже сейчас (недавно вышел подобный нетбук от китайской компании Menq). В Microsoft предполагают заменить Windows CE новой мобильной системой на базе Windows 7, которая сейчас в активной разработке. Возможно, эта ОС попадет и на процессоры ARM, раз уж они популярны у производителей смартфонов и коммуникаторов. А компания ARM анонсировала новые двухъядерные процессоры – Cortex A9. Итак, ситуация неоднозначна, и делать выводы рано; ждем появления первых массовых нетбуков на ARM.

Что было

» «**Переход на свободное программное обеспечение: реальные шаги**»

Такая конференция впервые состоялась 30 октября 2009 года в Красноярске. Целью ее было содействие местным организациям в переходе на СПО. Организаторы – Красноярский институт экономики Санкт-Петербургской академии управления и экономики и ООО «ИнФорм», при поддержке российского журнала Linux Format и группы российских пользователей *OpenOffice.org* MyOOo.ru. Половину из 50 участников конференции составили представители образовательных учреждений, четверть – ИТ-бизнеса, и еще четверть – других сфер бизнеса.

Особый интерес вызвали доклад об успехе внедрения GNU/Linux в Красноярском институте экономики и демонстрация приложений *Ekiga*, *Evolution*, *Moodle*, CAD, *Maxima*. Более половины участников высказали заинтересованность в использовании таких свободных программ, с которыми раньше они не были знакомы.

Google: Одним можно, другим нет?

В отношении поискового гиганта все чаще приходится слышать эпитет «Империя Зла». Насколько он оправдан?

Эту заметку я решил написать после раздачи лишних приглашений в сервис Google Wave на популярном ресурсе linux.org.ru. Меня потрясло огромное количество откликов в стиле: «Google не нужен!». Примечательно, что подобным образом русскоязычное Linux-сообщество (по крайней мере, та его часть, что зарегистрирована на LOR) реагирует на любую новость о корпорации. Есть ли основания называть Google новой Империей Зла? Попробуем разобраться.

Компания постоянно открывает и продукты, и технологии. Только в ноябре разработчики получили написанный на Java *Wave Federation Server*, который можно запускать как расширение *OpenFire*. Кроме того, Google публикует описание протокола Wave Federation Protocol для взаимодействия между серверами Волны (специалисты компании подчеркивают, что это не окончательная версия). И, разумеется, тестовый сервис для разработчиков WaveSandbox.com позволяет испытать возможности взаимодействия серверов.

Вскоре после этого представители компании заявляют о планах по открытию ключевых инструментов *Closure Tools*, используемых для разработки Google Mail, Google Docs и Google Maps. Среди других проектов хотелось бы выделить браузер *Chrome*, который делают на основе свободного *Chromium*. За примерами данной модели разработки далеко ходить не надо: Red Hat и Fedora, Novell и OpenSUSE – образцы плодотворного сотрудничества корпораций и сообщества. Но почему-то Red Hat и Novell это разрешается, а на Google объявили «охоту». А ведь «проприетарный» Google Chrome совершенно бесплатен, в отличие от продуктов той же Red Hat, поддержка которых стоит немало. В общем, на мой взгляд, имидж врага открытого сообщества Google получила абсолютно незаслуженно – на самом деле, сегодня это одна из крупнейших корпораций, занимающаяся разработкой открытого ПО! И тот факт, что на своих сервисах Google зарабатывает деньги, ничего не меняет – флагман движения, Red Hat, свободным ПО занимается тоже не из любви к искусству.

Gnome: Путем эволюции

Кнопка Пуск и традиционные одна-две панели уступят место более прогрессивному окружению, но не ранее сентября 2010 года.

Многие помнят, как выпускали KDE 4 – новая версия окружения рабочего стола была разработана практически с нуля. Можно долго спорить, насколько оправдан такой подход, но KDE «лихорадит» до сих пор. Кроме того, из-за экспериментов разработчиков проект потерял часть пользовательской аудитории.

Теперь на очереди GNOME 3 – в последней версии стал доступен новый интерфейс: *Gnome Shell* (стр. 7), который со временем заменит привычные пользователям панели. И хотя разработчики GNOME не пытаются по примеру конкурентов переписать всю среду сразу – они предпочли постепенное изменение, рабочий стол GNOME 3 отличается от предыдущих версий едва ли не сильнее, чем KDE 4 от KDE 3. Пока сложно сказать, чем все это кончится, но уже принято решение перенести релиз GNOME 3 на сентябрь 2010 года. За это время все разработки должны быть доведены до стабильного уровня. Будем надеяться, что в команде GNOME не станут увлекаться новым функционалом в ущерб стабильности, новый продукт будет полностью готовым к употреблению и не потребует доводки на протяжении целого года.



› Gnome Shell заменит собой традиционные панели рабочего стола.

Конференция выявила и консолидировала Linux-сообщество Красноярска. Анализ анкет участников показал, что 55 % из них уже используют GNU/Linux, а 27 % планируют начать его использование в ближайшем будущем. Не остался в стороне и актуальный на сегодня вопрос перехода школ на СПО. В ближайших планах организаторов – проведение семинара на тему внедрения СПО для директоров и учителей информатики школ Красноярска. linux.enisey.com

» Первый тур Всероссийской студенческой Олимпиады «IT-Планета 2009»

Более 6300 студентов со всей России из 570 учреждений среднего и высшего профессионального образования приняли участие в первом отборочном онлайн-туре, пройдя тесты на официальном сайте Олимпиады www.planet-it.ru.

Тесты разработали региональные ИТ-компании – партнеры «IT-Планеты», при поддержке Intel, Microsoft, D-Link, Oracle, 1C, LinuxCenter.Ru и Adobe Systems. Соорганизатор и генеральный партнер IT-Олимпиады в Южном федеральном округе – ОАО «Южная телекоммуникационная компания», а информационными партнерами в Российской Федерации в 2009 году стали журналы «CRN/RE» и «Linux Format».

Прошедших во второй тур внесут в базу данных перспективных студентов, рекомендованных Центральным оргкомитетом Олимпиады региональным работодателям, кадровым агентствам и органам государственной власти, что увеличит их шансы на трудоустройство и удачный карьерный старт.

«В ЮФО IT-Олимпиада проходит уже не впервые, но приятно удивил огромный интерес, проявленный к проекту его участниками, – говорит Анна Сергиенко, зам. председателя Оргкомитета по ЮФО IT-Олимпиады. – Замечательно, что талантливым и активным молодым специалистам помогут в профессиональном самоопределении и трудоустройстве». LXF



Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов



АЛЕКСЕЙ ФЕДОРЧУК
Его слабости – mass storage, разметка диска и файловые системы.

GNOME Shell: всё для блага человека

GNOME Shell – новый способ взаимодействия пользователя с рабочим столом GNOME, предлагаемый разработчиками как революционный и «пользительный». Предполагается, что он станет стандартным в грядущей 3-й версии этой среды, но уже сейчас его можно без труда включить в GNOME 2.28 (например, в Fedora 12).

Революционность *GNOME Shell* – в наличии двух принципиально разных режимов: «оверлейного», котором можно только запускать приложения и открывать документы, причем делать это абсолютно единообразно, и «рабочего», в котором, собственно, и выполняется работа.

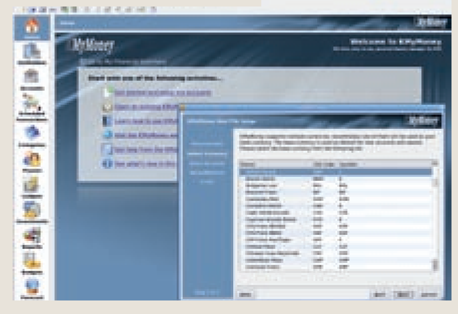
Переключение между режимами выполняется как с помощью мыши, так и очень удобными горячими клавишами, чем и определяется одна сторона его «пользительности». Вторая же сторона – разгруженный интерфейс, оставляющий максимум простора как для манипуляции приложениями и файлами, так и непосредственно для работы. Это не столь критично на больших экранах настольных ПК, но окажется востребованным на крошечных дисплеях нетбуков, а возможно, даже и MID.

В современном виде *GNOME Shell* еще не очень пригоден для повседневного применения, в частности, из-за скудости настроек. Но опробовать его потенциал можно уже сейчас. Мне – понравилось...

alv@posix.ru

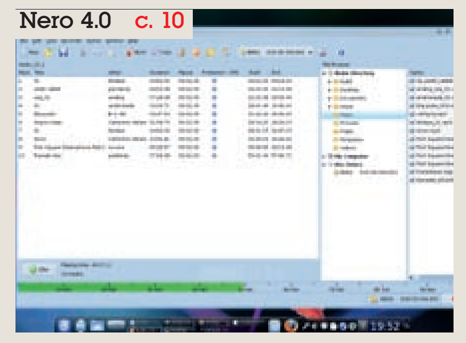
Сегодня мы рассматриваем:

KMyMoney 1.0 с. 11



› Непрерывно можно смотреть на три вещи: огонь, воду и на то, куда утекают ваши денежки.

Nero 4.0 с. 10



› *Nero* – превосходное приложение для Windows, но не потеряло ли оно что-нибудь при переезде под Linux?

Наш вердикт: Пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибалльной шкале (0 – низшая оценка, 10 – высшая). Как правило, учитываются функциональность, производительность, простота использования и цена, а для бесплатных программ – еще и документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.



Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но если разработчики рекомендуют *Autopackage*, мы следуем этому совету.

LINUX FORMAT Вердикт

Google Earth

Разработчики: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Оправданность цены	9/10

» Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряюще практичная программа.

Рейтинг 9/10

Slackware 13.0



Истоки его теряются во тьме веков, а он и ныне на коне. Кто или что это? **Ник Вейч** усмотрел в себе и Slackware много общего.

Вкратце

» **Популярный дистрибутив без фокусов. Ровесник Debian.**

Ключевое ПО

- » Ядро 2.6.29.6
- » KDE 4.2.4
- » Xfce 4.6.1
- » Firefox 3.5.2
- » Thunderbird 2.0.0.23
- » X.org 1.6.3

Slackware – один из старейших дистрибутивов Linux: он на месяц опередил Debian в далеком 1993-м; но не самый старый. Первые его версии были основаны на SLS, который по праву претендует на звание «прародителя» Linux-дистрибутивов. Благодаря тщательности своей компоновки, Slackware быстро обрел последователей. С тех пор в идеологии дистрибутива мало что изменилось, и ведущий разработчик Патрик Фолькердинг [Patrick Volkerding] по-прежнему героически трудится, ежегодно выпуская новую версию.

Вы поймете, подойдет вам Slackware или нет, вскоре после запуска установочного скрипта. Не смогли ответить на заданные вопросы или разбить диск из командной строки – считайте, что вы не готовы к работе с дистрибутивом, даже при успешной инсталляции.

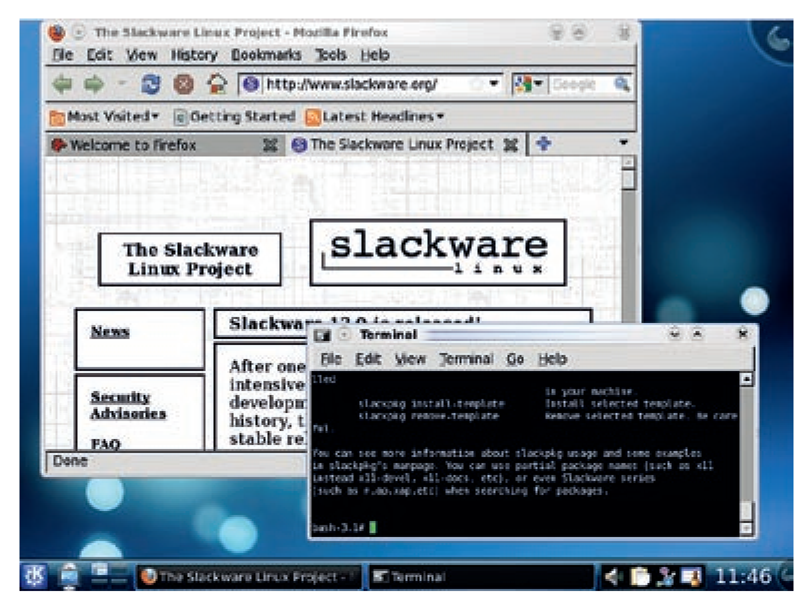
Это не пижонство текстового мачо, и дистрибутив вовсе не лезет из кожи вон, чтобы выбить вас из колеи: просто здесь нет двадцати пяти слоев ПО, подстилающих изнеживающий графический интерфейс.

> «Самая радикальная перемена – новый формат пакетов.»

После установки версия 13.0 обнаруживает немало новшеств – по крайней мере, для Slackware. Прочие дистрибутивы уже переключаются на KDE 4.3, а Slackware выбрал KDE 4.2.4. Предыдущая версия выходила с *KDE 3* – так что шаг сделан широкий, и мы ждем неоднозначной реакции на него со стороны более консервативных пользователей дистрибутива. Впрочем, если KDE 4 вам не по вкусу, всегда можно переключиться на *Xfce* или *Fluxbox*. Gnome не входит в стандартный комплект: обычно его поставляют сторонние разработчики вроде Gware (www.gware.org) или Dropline (www.droplinegnome.org).



Помимо перехода на KDE, теперь стало проще управляться с *X*-сервером благодаря включению нового *X.org*: он почти всегда способен сам написать для себя конфигурационный файл. Но имейте в виду: заставить сервер работать с проприетарными драйверами – не самая тривиальная задача.



› **Вот оно, будущее! Ну, хорошо: последняя стабильная версия будущего. Стандартный рабочий стол теперь KDE 4.2.4, но доступны и другие.**

Не расслабляться!

Но самое радикальное изменение – это, пожалуй, новый формат пакетов. Последние 16 лет основным способом распространения Slackware были tar-архивы. Они по-прежнему в ходу, но дистрибутив теперь базируется на пакетах с расширением TXZ. По сути, это «в лоб» и «по лбу». Просто упаковочные инструменты обновили методы сжатия, включив новый стандарт LZMA (формат архиватора с открытым кодом *7-Zip*). Причина изменения проста – свободное место. Файлы LZMA примерно на 5 % меньше zip, поэтому на DVD теперь можно втиснуть больше ПО (хотя на *OpenOffice.org* по-прежнему не хватило).

В повседневном использовании Slackware хорош для тех, кто умеет его настроить, но не годится для новичков. В некотором смысле обманчиво само название дистрибутива [slack = слабость, вялость]: особо с ним не расслабишься. И тут есть свой резон: Slackware – один из наиболее «юниксоидных» вариантов Linux, бережно соблюдающий лучшие традиции. Помните, сколько раз вы вручную рихтовали конфигурационные файлы или **fstab** после вмешательства *Yast* и ему подобных (ОС, видите ли, возомнила себя умнее вас)? В Slackware подобное в принципе невозможно. Настраивать систему с помощью *Vim* не так легко, как в графическом приложении, зато вы ощущаете себя хозяином положения и полностью контролируете процесс.

У Slackware немало поклонников среди пользователей ПК и серверов, и новый релиз не дает оснований думать, что версии Slackware 14.0 не будет. LXF



LINUX FORMAT **Вердикт**

Slackware 13.0

Разработчик: Патрик Дж. Фолькердинг
Сайт: www.slackware.org
Цена: бесплатно на условиях GPL

Функциональность	6/10
Производительность	9/10
Простота использования	5/10
Документация	7/10

» Стабилен, безопасен и прост. Если у вас достаточно времени и знаний для правильной настройки – вы его полюбите.

Рейтинг 7/10

Parallels Desktop 4

Как всякий ценитель виртуализации, **Маянк Шарма** обрадовался появлению нового клиента Parallels для Linux. И, на свою голову, его установил...

Вкратце

» **Настольный виртуализационный клиент с хорошей родословной. См. также:** *VirtualBox*, *Win4Lin* **или** *VMware Workstation*.

Завоевав виртуализационное пространство на Mac, *Parallels* ввязался в битву с *VMware* за клиент для Windows и Linux. Однако в Linux битва оказалась весьма свирепой: *Parallels* приходится соперничать и с проприетарными (вроде *VMware Workstation*), и со свободно-бесплатными (см. *VirtualBox* от Sun) решениями. Так как же смотрится *Parallels Desktop 4 for Linux (PD4)* на поле брани?

PD4 способен обслуживать как 32-, так и 64-битные гостевые ОС и наделять свою виртуальную машину (ВМ) несколькими процессорами, общим числом до восьми. К ней можно подключить до 16 сетевых карт, до 8 ГБ виртуальной памяти и до 2 ТБ виртуального дискового пространства.

К сожалению, *PD4* не может соединяться с устройствами FireWire и не работает с 3D, хотя у конкурентов всё это имеется (пускай и в бета-состоянии).

> «PD4 поддерживает аппаратные технологии Intel VT-X и AMD-V.»

Есть и аппаратный недостаток: *PD4* поддерживает аппаратные технологии виртуализации Intel VT-X и AMD-V, но, в отличие от *Workstation* и *VirtualBox*, без этих расширений не работает. Хотя процессоры с их поддержкой и недороги, но обязательность их применения оставит за бортом многих пользователей: переходя на двухъядерный процессор, не каждый удосужится прочесть приписку мелким шрифтом.



› *Parallels* вышел из Mac, и... ударил на Linux в грязь лицом.

Но наряду с этими недостатками, у *PD4* есть и достоинства: например, набор приложений для создания и текущего обслуживания ВМ.

Одна из таких программ, Parallels Transporter, предназначена для переноса физической инсталляции на ВМ. Для кого-то это пустой звук, но при установке по сети такую возможность трудно переоценить. Кроме того, данное приложение можно использовать в *PD4* для портирования ВМ (вместе с виртуальными дисками), созданных другими приложениями, например, *Workstation*.

Инструментарий

Image Tool, инструмент с приятным интерфейсом, поможет изменить размеры и свойства (тип и формат) виртуальных дисков. Parallels Compressor служит для ухода за дисками: дефрагментации и сжатия за счет оптимизации свободного пространства. Очень удобно для фиксации объемов ВМ (например, тех, что работают с унаследованными приложениями).

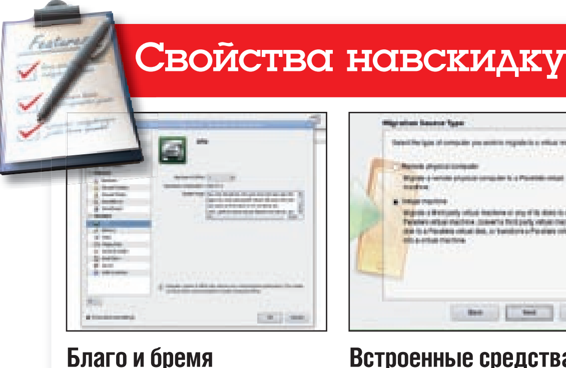
И всё же *Parallels* оставляет ощущение однобокости. Два инструмента из названных работают только с Windows: Transporter импортирует лишь «гостей» от Microsoft, а Compressor признаёт только виртуальные машины, запущенные в Windows.

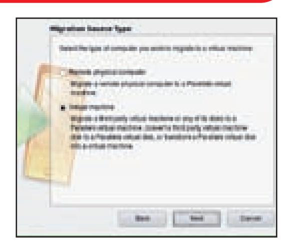
Есть и другие симптомы приверженности к Windows. Так, Express Install для бесперебойной установки требует заранее указать номер Windows-лицензии. Тем же свойством отличаются *Workstation* и *Win4Lin*.

Допускается вынос приложений из гостевой Windows на хост. Такая функция тоже есть у *Workstation*, *VirtualBox* (оба клиента могут «вытягивать» приложения из Windows- и Linux-гостей) и *Win4Lin*.

Если отвлечься от того факта, что *PD4* может работать с гостевыми Linux-ВМ (что несложно, учитывая полное отсутствие Linux-специфики), то данное ПО можно охарактеризовать как нишевый продукт, предназначенный для работы с гостевыми Windows на Linux-хостах. Сравнивать его следует скорее с *Win4Lin*, чем с полноценными виртуализационными решениями для Linux. LXF

Свойства навскидку





Благо и бремя
Без виртуализационных расширений процессоров скорость в *PD4* не повысить.

Встроенные средства
Благодаря комплекту инструментов управления виртуальные машины работают как часы.



LINUX FORMAT Вердикт

Parallels Desktop 4 for Linux

Разработчик: Parallels
Сайт: www.parallels.com
Цена: $79,99

Функциональность	5/10
Производительность	6/10
Простота использования	7/10
Оправданность цены	5/10

» Фурора в мире виртуализации не произведет, но для работы с гостевыми Windows сгодится.

Рейтинг 6/10

Nero 4.0

Стряхнув пыль с пишущего привода, **Грэм Моррисон** испытал новый коммерческий дископрожигатель. Ну все как по маслу!

Вкратце

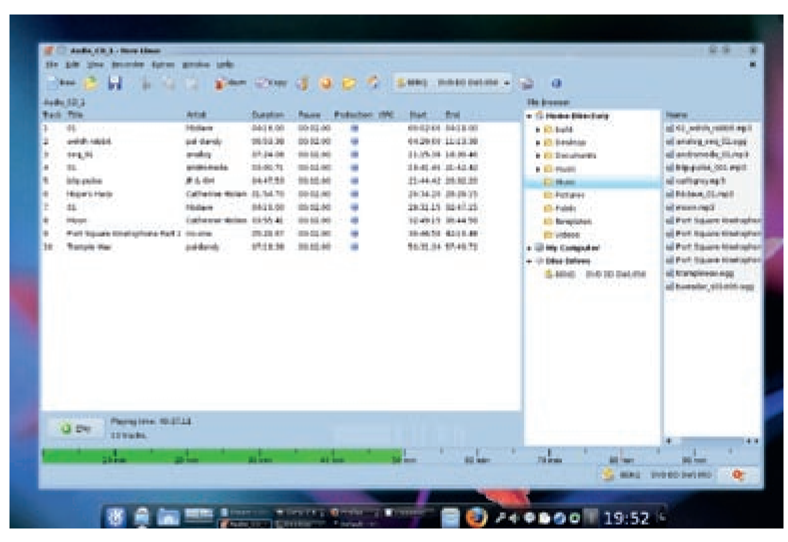
» Инструмент записи CD, DVD и Blu-ray с помощью мастера или специализированного файлового менеджера. См. также *K3b* (функциональность) или *Brasero* (интерфейс мастера).

Несмотря на наличие двух отличных открытых приложений для прожига CD и DVD (речь идет о *Brasero* и *K3b*), *Nero* – коммерческий и проприетарный – упорно продолжает «окучивать» рынок Linux. И это радует (версия 4, во всяком случае).

Впервые *Nero* для Linux функциональным изобилием напоминает *Nero* для Windows. Наряду с двумя крупными новшествами (мастером создания дисков и 64-битным вариантом), мы заметили множество более мелких, но исключительно полезных добавок. Мы устанавливали 64-битный Deb-пакет на Linux Mint и Kubuntu; доступны версии для Fedora и OpenSUSE. После установки вы найдете в меню программ два новых пункта. Первый – мастер создания дисков, по прозвищу *Nero Express* (очень близкий к Windows-аналогу). Он занимает место файлового менеджера, традиционного для большинства дискозаписывающих программ, и проводит пользователя сквозь различные стадии записи.

> «Форматы преобразуются так быстро, что даже не верится.»

Работать с аудиодисками в *Nero* – одно удовольствие. Можно добавлять списки воспроизведения M3U или PLS, а также сами файлы MP3 и Ogg Vorbis. В новой версии поддерживаются форматы AIFF и Musepack, а паузы между дорожками в финальной версии можно убирать. Мы нашли отображение дорожек и индикатор свободного места чуть более удобными, чем в *K3b*, а от ограниченного интерфейса *Brasero* отрыв весьма значительный. Форматы перекодируются очень быстро: мы даже не были уверены, что это вообще произошло, пока не услышали музыку с записанного диска.



› Для обычной работы с аудиофайлами при записи CD *Nero* — лучшее приложение из всех, нами виденных.

Второй мастер, для видео-DVD, действует примерно так же, но любые функции преобразования форматов отсутствуют напрочь, то есть нужно самостоятельно подготовить DVD-совместимые файлы к записи. Для Linux-пользователей это минус, и возможность кодирования видеофайлов хотелось бы видеть уже в следующей версии. Наконец, с помощью мастера можно создавать образы дисков, а также записывать любые имеющиеся образы на носители.

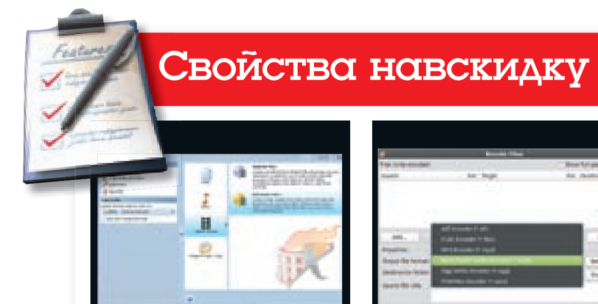
Грузим Linux

Прямо в основном интерфейсе приложения можно создавать загрузочные Linux-диски с *Isolinux* или удобные DR-DOS для «ремонта» Windows-систем. Кроме того, *Nero* может кодировать аудиофайлы в форматы MP3, MP4, Ogg и Flac и получать названия дорожек с http://freedb.org.

Если вам нужен более тонкий контроль над процессом создания диска, то к вашим услугам интерфейс файлового менеджера – надежное проверенное решение. Возможно управление процессом перепрожига (если вам это надо) и запись на диски CD, DVD5, DVD9 и Blu-ray, хотя последний вариант мы протестировать не смогли.

Сам процесс записи безупречен, чего и следовало ожидать от приложения такого уровня. Единственная проблема – нехватка документации: если нет доступа в Интернет, то читать вообще нечего, да и онлайн-справка небогата. Но если брать в целом, то у *Nero* превосходное соотношение «цена–качество» для коммерческого пакета с набором средств аудио-кодирования и непревзойденными возможностями записи. LXF

Свойства навскидку



Мастер
С новым мастером диски можно записывать парой щелчков мыши – здорово экономит время, если их много.

Загрузка Linux
Для создания загрузочных носителей *Nero* может использовать *Isolinux* или старый добрый DOS.



Linux Format Вердикт

Nero 4.0

Разработчик: Nero AG
Сайт: www.nero.com
Цена: € 19.99

Функциональность	9/10
Производительность	9/10
Простота использования	8/10
Оправданность цены	9/10

» Nero постоянно совершенствуется: этак он станет нашим любимым инструментом для прожига дисков.

Рейтинг 9/10

KMyMoney 1.0



Отчаявшись разобраться с хроническим завалом транзакций, **Маянк Шарма** разыскал-таки для этого бухгалтерское приложение.

Вкратце

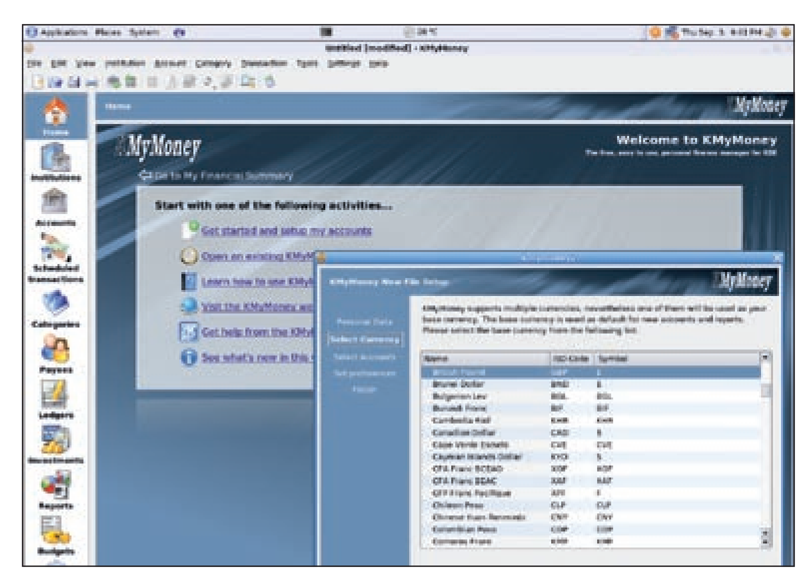
» **Бухгалтерское приложение для ведения персонального бюджета. Среди альтернатив:** *GnuCash*, *Grisbi* и *Moneydance*.

Персональные финансовые менеджеры помогают поддерживать иллюзию контроля над нашими деньгами. Если вы ищете средство учета ежемесячных расходов, то *KMyMoney*, как раз дошедший до взрослой версии 1.0, выглядит многообещающе.

Впрочем, следует вас предостеречь: хотя *KMyMoney* и не так сложен, как *GnuCash*, выучиться работе с ним будет ох как непросто. *KMyMoney* помогает вести учет банковских и кредитных счетов, займов, инвестиций и т.п. Приложение целиком сфокусировано на персональных финансах: в нем нет средств управления счетами, выплатой зарплаты и других сугубо деловых функций.

Прежде чем запускать *KMyMoney* впервые, не пожалейте времени на изучение руководства пользователя, подробного и богато иллюстрированного. Это поможет вам пройти через начальную настройку – при первом знакомстве вас забросают неудобопонятными вопросами, да еще не все они касаются вас: например, потребуют код банка или идентификационный код. А они актуальны только для ведения интернет-банкинга, и в другой ситуации можно обойтись и без них.

> «Модули приложения отлично увязаны друг с другом.»



› *KMyMoney* **проследит за вашими вложениями на биржах, паевых фондах и т.п., и автоматически добудет последние цены с finance.yahoo.com.**

Одно из замечательных свойств *KMyMoney* – виртуозность учета регулярных доходов и расходов. Можно добавить нескольких получателей платежей и составить график выплат в соответствии с параметрами автокредитования, предоставляемого вашим банком. Разнообразные модули приложения отлично увязаны друг с другом, так что добавить получателя можно и при составлении графика. Облегчая вашу работу, *KMyMoney* автоматизирует многие повторяющиеся функции: например, увеличивает номера заполняемых подряд чеков. При желании автоматику всегда можно отключить или поправить.

Выписки и графики

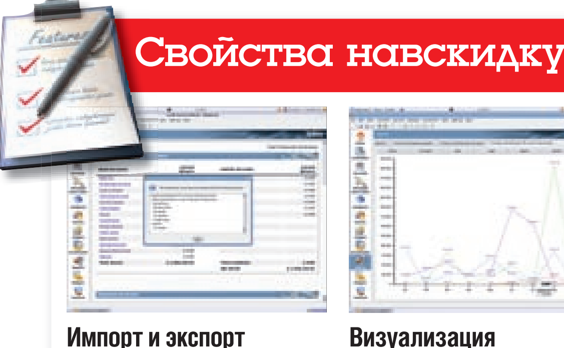
Получив выписку о состоянии счета от банка, эмитента кредитной карты или заимодателя, вы можете сверить ее с бюджетом *KMyMoney*. Возможен импорт электронной выписки, что еще более упрощает сверку.

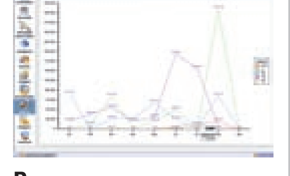
В последней версии *KMyMoney* появилась впечатляющая функция составления отчетов и диаграмм – она позволяет наглядно оценить ваше финансовое положение в графической форме. Приложение оснащено модулем прогнозирования финансового состояния на некоторый период в будущем. Прогноз основывается на обработке прошлых транзакций или на анализе графика выплат.

Еще одно новшество последней версии – возможность переноса бухгалтерии в базу данных, что упрощает экспорт и делает данные доступными для других приложений. В *KMyMoney* используется модуль SQL от *Qt*, который предоставляет драйверы для самых различных СУБД. Среди них *MySQL*, *PostgreSQL*, *SQLite*, *Oracle*, *IBM DB2* и др.

Реальный недостаток только один – *KMyMoney 1.0* пока несовместим с KDE 4.x, но порт уже разрабатывается. LXF

Свойства навскидку





Импорт и экспорт

Кроме стандартных форматов (QIF и OXF), добавлена поддержка популярных баз данных.

Визуализация

Удобочитаемые отчеты и графики помогут отследить состояние различных параметров (чистый капитал, суммарные доходы и расходы) в любой момент.

Linux Format Вердикт

KMyMoney 1.0

Разработчик: The KMyMoney team
Сайт: http://kmymoney2.sourceforge.net
Цена: бесплатно на условиях GPL

Функциональность	8/10
Производительность	8/10
Простота использования	7/10
Документация	8/10

» Комплексное средство управления личными финансами, подкрепленное богатой документацией и активным сообществом.

Рейтинг 8/10

Zend Studio 7.0

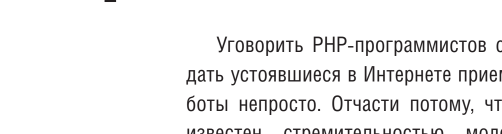
Стоит ли набирать код PHP в *Gedit*, имея многофункциональный Zend Studio? **Пол Хадсон** взвешивает «за» и «против».

Вкратце

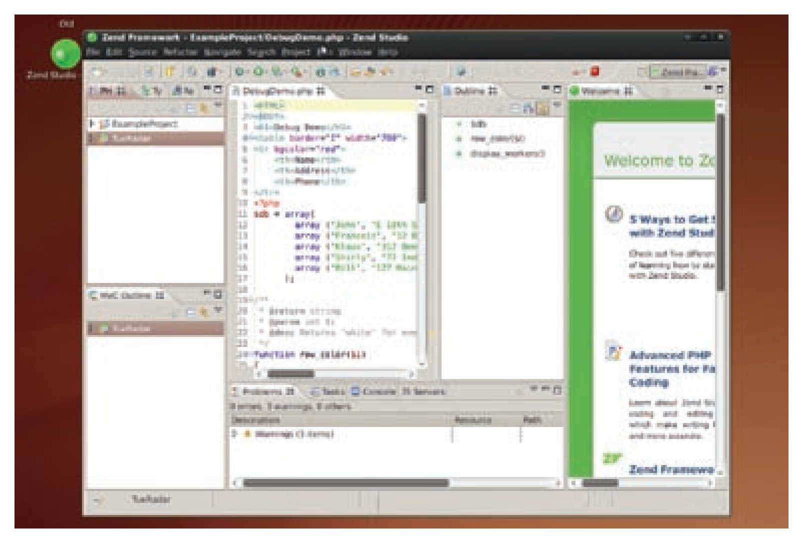
» **Среда разработки PHP, инструменты и функции которой реализованы через... э-э, не лучшее место. См. также: *Komodo* от ActiveState.**

PHP долгое время считали языком программирования для баловства, и не без оснований. Нехватка свойств вроде пространств имен или анонимных функций, наплевательский подход к безопасности и пренебрежение тестовыми процедурами часто делали PHP темой анекдотов. Компания Zend, претендующая на роль «покровителя» PHP, затеяла устранение недостатков и, не копируя на скорости Perl 6, пытается исправить все чохом.

Zend Studio по-прежнему базируется на платформе *Eclipse*, но в версии 7.0 немало новшеств: это поддержка PHP 5.3, куда вошли пространства имен, анонимные функции, а позднее – статическое связывание, расширения из PHP Extensions Community Library и пр. Это самое значительное из всех обновлений для серии 5.x: приятно видеть, что Zend подтолкнула снабжение программистов лучшими инструментами.



Уговорить PHP-программистов соблюдать устоявшиеся в Интернете приемы работы непросто. Отчасти потому, что PHP известен стремительностью моделирования, тогда как общепринятые процедуры далеко не просты и быстротой не отличаются. Решение Zend? Написать совершенно новый каркас и бросить вызов Ruby On Rails с MVC, объектной ориентацией, высокопроизводительным кэшированием, культурой модульного тестирования, стандартами программирования и богатой документацией. В состав версии 7.0 входит новейшая версия каркаса, 1.9, а с ней улучшенные API RSS, LDAP и очередей сообщений.



» **Для создания вашего первого проекта Zend Framework по умолчанию выкатит вам ЭТО. Вот и мы не знаем, чем думали разработчики...**

PHP по-взрослому

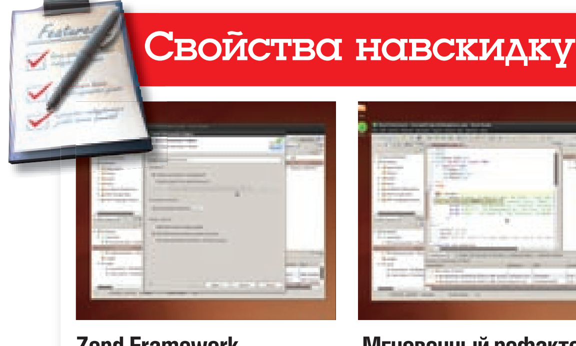
Zend Studio 7.0 отметает всякие сомнения в пригодности PHP к использованию на предприятии; чего же еще не хватает? Проблема в том, что PHP 5.3 и Zend Framework, два столпа нового ПО, бесплатны и доступны любому, у кого есть *Gedit* и подключение к Интернету.

Zend Studio пытается создать добавочную ценность за счет сбора всех функций воедино, для удобной и надежной разработки ПО индустриального уровня. То есть вы можете установить приложение на сервер, дистанционно вылавливать и ликвидировать ошибки и анализировать серверные протоколы в случае сбоя. В области отладки *Zend Studio* по-прежнему нет равных, а теперь эта функция подкреплена еще и развитым редактированием кода.

И все-таки сомнения остаются. Частично – из-за цены: программист-одиночка, работающий над мелкими проектами, еще подумает о приобретении лицензии, но для группы разработчиков плата за все машины и всех участников влетит в копеечку. Вряд ли найдется компания – любитель добровольно раздувать свой ИТ-бюджет. Поэтому нас ничуть не удивит, если один-два программиста из группы будут пользоваться *Zend Studio*, а остальные члены команды предпочтут нечто подешевле (если не вовсе бесплатное).

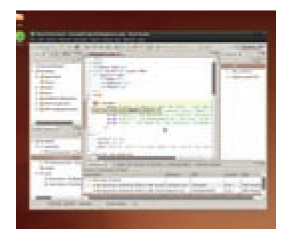
Кроме того, *Zend Studio* оставляет ощущение незавершенности. Стандартный проект Zend Framework выдал ошибку (не определена переменная), контекстное меню растянулось на 30 пунктов, а текст, предлагаемый для автозавершения, зачастую не вмещается в рамку подсказки. Работать в такой среде не очень-то приятно, и единственный способ исправить положение – реформы интерфейса. LXF

Свойства навскидку



Zend Framework

Поддержка Zend Framework 1.9 включена по умолчанию: Dojo активируется одним щелчком мыши.



Мгновенный рефакторинг

Рефакторинг «на ходу» позволяет вносить безотлагательные изменения – одна из новых функций.

LINUX FORMAT Вердикт

Zend Studio 7.0

Разработчик: Zend
Сайт: www.zend.com
Цена: € 399

Функциональность	10/10
Производительность	8/10
Простота использования	5/10
Оправданность цены	5/10

» Богатейшая среда разработки PHP, но дубовый интерфейс обесценивает ее достоинства.



7/10

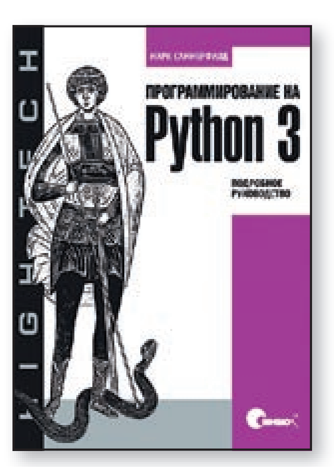
Программирование на Python 3. Подробное руководство

В свое время о Py3k говорили много и разное. **Валентин Синицын** решил узнать, стоит ли знакомиться с новой версией по этому учебнику.

В **LXF115** мы жаловались, что книгоиздание никак не угонится за быстро меняющимся миром ИТ. Тогда мы имели в виду учебник по Python 2.5, вышедший в эру 3.0. Наш клич был услышан, и эта книга появилось даже раньше, чем Python 3.0 реально проник на наши компьютеры.

Ее автор, Марк Соммерфильд [Mark Summerfield], известен отечественному читателю учебниками по *Qt*. «Программирование на Python 3» – обстоятельное введение в новый во всех смыслах язык программирования и его стандартную библиотеку. Оно насчитывает 13 глав; практически каждая из них завершается парой крупных примеров и заданиями для самостоятельного решения. Еще изюминка – перекрестные ссылки, указывающие, в каком разделе можно получить углубленные сведения по текущей теме. Самые нетерпеливые могут ограничиться первой главой, содержащей достаточно, чтобы начать писать свои программы. Главы 2–6 раскрывают синтаксис (от типов данных и управляющих конструкций до объектно-ориентированного программирования) более подробно. В главах 7–12 рассматривается работа с файлами, многопоточное программирование, сети, взаимодействие с СУБД и регулярные выражения. Наконец, последняя глава посвящена созданию графического интерфейса с использованием *Tk*. Разумеется, он есть везде, и без *Tk* не обходится практически ни одна серьезная книга по Python, но нам было бы интереснее увидеть более распространенный инструментарий, скажем, *Qt*. Соммерфильд написал на эту тему целую книгу, но на русском языке она пока не вышла.

Нам случалось слышать нарекания на качество перевода – в отличие от верстки, особых проблем здесь мы не заметили. А вот чего, на наш взгляд, действительно не хватает – это быстрого введения в Python 3 для знакомых с 2.x. Оно не заняло бы много места, но расширило бы читательскую аудиторию от новичков Python до ветеранов.



› 608 страниц о Python 3.0 — не слишком ли много для «простого языка, идеального для первого знакомства»?



Программирование на Python 3. Подробное руководство

Автор: Марк Саммерфильд
Издательство: Символ-Плюс/Addison-Wesley
ISBN: 978—5—93286—161—5
Цена: 750 руб.
Объем: 608 стр.

» Хороший, но местами чересчур обстоятельный способ разобраться в Python 3 — опытный разработчик может заскучать.

Рейтинг 9/10

Django. Разработка веб-приложений на Python

Давненько не писавший ничего для Web **Валентин Синицын** узнал, что современная web-разработка – это не только PHP и RoR.

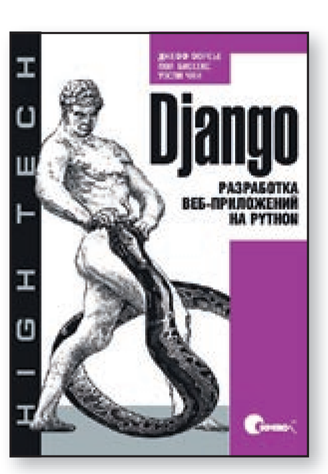
Создать сайт можно тремя способами: взять готовый (обычно бесплатно), развернуть CMS-систему или же реализовать идею с нуля, желательно – с использованием web-каркаса. Об одном из них, *Django*, и рассказывает эта книга. Не считая серии учебников, выходивших в **LXF105–109**, она – единственное упоминание о *Django* в русскоязычной печати, что само по себе достаточный повод взглянуть на нее поближе.

Содержание можно поделить на три части. Сперва идет введение в Python (2.6) с прицелом на *Django* – хороший способ освежить в памяти язык, если вы его подзабыли. Для опытных разработчиков оно может стать экспресс-курсом по Python, как и следующая глава, описывающая создание простейшего блога – по *Django*.

Вторая часть посвящена составляющим архитектуры *Django 1.0*. Модели, представления, шаблоны, URLConf и запросы – все это объясняется подробно и обстоятельно; жаль, ничего не говорится о локализации (впрочем, для переводного издания это не удивительно). Третья часть – примеры «настоящих» web-приложений: фотогалереи, еще одного блога (с Ajax), CMS-системы и Patebin. Книга не является пособием по web-дизайну и клиентским технологиям, но создаваемые авторами шаблоны не примитивны и дают представление об использовании *Django* в реальных условиях.

В завершение обсуждаются вспомогательные вопросы: установка *Django*, работа в командной строке и интеграция с Google App Engine. Лично мы бы предпочли прочесть побольше про GAE и поменьше – про командные оболочки, но зато такой подход дает читателю необходимый минимум знаний для работы на Unix-сервере в качестве обычного пользователя.

Текст написан хорошо и легко читается; не хватает разве что нумерации строк в крупных примерах. Свободный Django Book (www.djangobook.com) еще не достиг версии 1.0, и данная книга стоит внимания, если вы беретесь за web-приложения. LXF



› «Итеративный» подход к изложению материала позволит как познакомиться с *Django*, так и узнать о нем в подробностях.



Django. Разработка веб-приложений на Python

Авторы: Дж. Форсье, П. Биссекс, У. Чан
Издательство: Символ-Плюс/Addison-Wesley
ISBN: 978-5-93286-167-7
Цена: 490 руб.
Объем: 456 стр.

» Должно понравиться как начинающим web-программистам, так и пользователям других web-каркасов.

Рейтинг 8/10

Книги для данного обзора были предоставлены интернет-магазином Books.Ru.

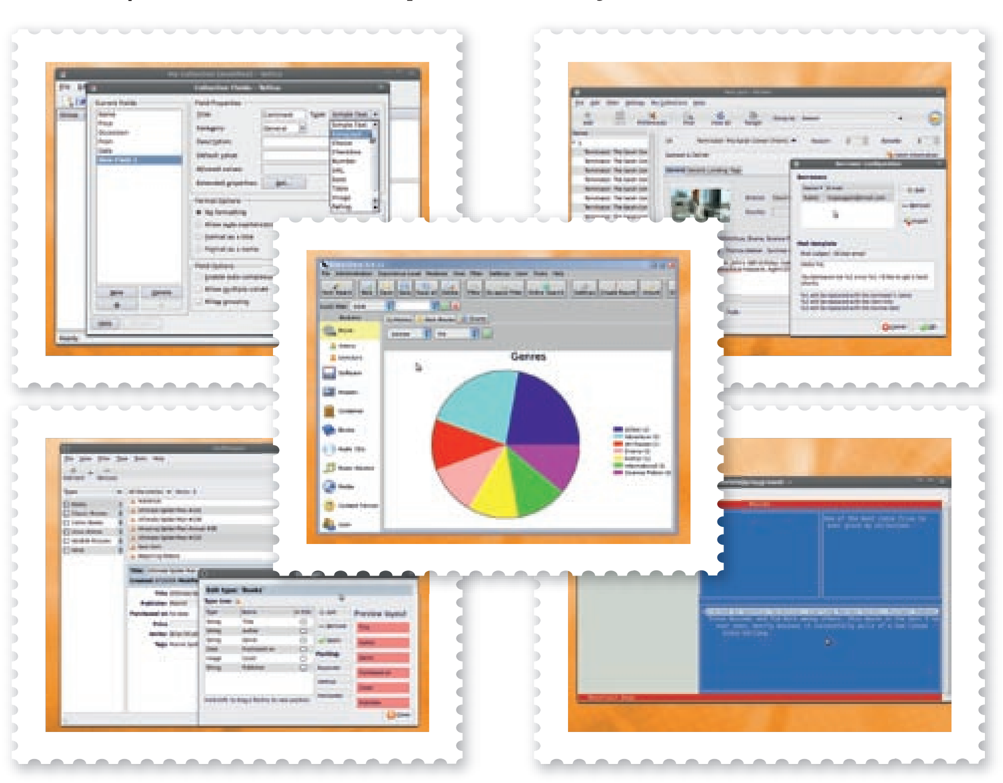


Каждый месяц мы сравниваем тысячи программ – а вы можете отдыхать!

LINUX FORMAT на DVD

Менеджеры коллекций

У каждого найдется коллекция, требующая заботы – будь то DVD, музыка или марки. **Шашанк Шарма** исследует ПО, подходящее для этого дела.



Про наш тест...

Существует масса программ, способных следить за вашими коллекциями книг или музыки, но среди них довольно мало таких, которые предлагают поддержку разнородных элементов, что нас интересует сейчас. Менеджеры коллекций создаются с одной целью: помочь вам управлять вашей коллекцией, что бы ни было ее предметом. Хотя многим из нас вполне хватает текстового редактора, грамотный менеджер коллекций со встроенной поддержкой различных типов элементов облегчит задачу в разы.

Итак, нужен самый подходящий ассистент. Для его выбора мы разузнаем, можно ли осуществлять поиск элементов и находить сведения о них в Интернете, упрощают ли вашу жизнь функции, включенные в программу, и можно ли, при необходимости, экспортировать данные.

Наш выбор

Коллекционирование заложено в природе человека. Вещи, которые человек копит, со временем меняются, но сам процесс не прекращается. Мы собираем синяки и царапины в возрасте семи лет, карточки – подростками, и, хотя в этом редко кто признается, те красивые обертки с бесплатными татуировками тоже.

Минуту подумав, вы, вероятно, вспомните не одну коллекцию, продолжающую расти благодаря вашим усилиям. Среди них – книги, фильмы, монеты, карты, вина или сувениры из ресторанчиков, где вы побывали; у вас найдется уйма различных коллекций, требующих структурирования.

И тут нам помогут менеджеры коллекций. В наши дни большинство из них имеет красивый интерфейс, способный получать информацию через Интернет, чтобы помочь вам каталогизировать свою коллекцию. Кроме того, они часто позволяют создавать бирки-тэги для экспонатов коллекций, осуществлять поиск по коллекциям и даже экспортировать информацию в другую систему.

Пока вы не окончательно избалованы выбором, разделим доступные инструменты на два лагеря. Существуют всеобъемлющие менеджеры с целым спектром проверенных встроенных шаблонов и поддержкой определенных типов коллекций, и их легковесные конкуренты, управляющие полями данных, ассоциированными с каждой отдельной коллекцией, сведения о которой вы вводите.

Однако шик графического интерфейса еще не гарантирует, что эти программы эффективнее классической амбарной книги в кожаном переплете или современного текстового редактора. Так кому, если вообще какому-то, из подобных приложений можно вверить вашу сокровищницу? И что они дают сверх обычных записей на бумаге?

> «У вас найдется уйма коллекций, требующих структурирования.»

Tellico

Старичок все еще резво тянет.

Tellico написан для KDE, но чувствует себя как дома и в Gnome. Это, пожалуй, старейший менеджер коллекций из тех, что по-прежнему в седле сегодня. Он поставляется со встроенной поддержкой различных типов коллекций: книги, фильмы, музыка, видеоигры, комиксы, монеты, марки, билетики, вино, настольные игры… и многое другое. Причем вместо использования баз данных для хранения всей этой информации, *Tellico* полагается на формат XML.

Web-страница *Tellico* предлагает подробное иллюстрированное руководство в дополнение к богатой документации, но недостаток обширной встроенной поддержки – присущий ей не в меру напористый интерфейс, хотя это и отражение функций программы. Инвентаризуя нашу коллекцию комиксов, мы, честно говоря, ленимся вводить дату покупки, и малость раздражает, что *Tellico* к нам с этим пристает.

Даже если заполнение всех, а то и большинства полей не обязательно, результат не ласкает глаз. Использованные диалоговые окна забиты до отказа, а незаполненные поля уродливо зияют пустотой.

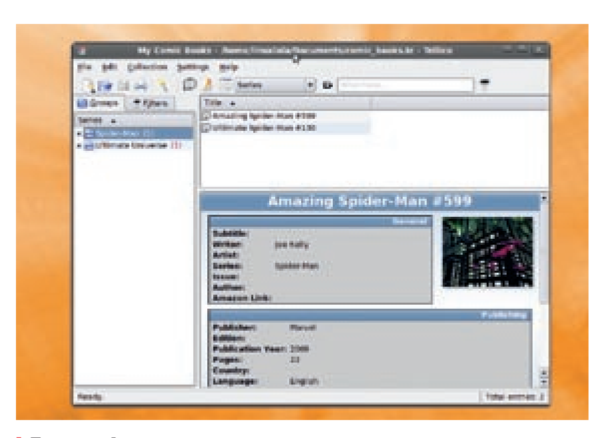
Правда, несмотря на неудачный отправной пункт, состав полей, связанных с каждым из встроенных типов коллекций, все-таки контролируется. Можно также менять параметры существующих полей. Допустим, вам незачем держать поле Дата покупки [Purchase Date] ни для одной из ваших коллекций; тогда в индивидуальном порядке отредактируйте и удалите его. Кроме того, в любой из шаблонов можно добавить новые элементы, если вы ощутите недостачу – например, текстового поля, описывающего ваши эмоции по отношению к данной вещи из коллекции.

Отслеживание займов

Tellico также является эффективным средством для приглядывания за коллекцией, поскольку он умеет отмечать, не одолжили ли вы вещь еще кому-либо. Отчеты о коллекции, пригодные для распечатки или, при желании, экспорта в HTML, создаются быстро, и для них есть различные шаблоны, в том числе Просмотр займов [Loan view].

> «Tellico поддерживает массу интернет-источников.»

Наконец, *Tellico* поддерживает массу различных интернет-источников, где добывается информация об элементах вашей коллекции. К ним относятся IMDB; ISBNdb (онлайновая база данных о книгах); CrossRef для научных статей и библиографических текстов, а также серверы SRU, применяемые многими библиотеками для обеспечения доступа к своим каталогам. В качестве бонуса, можно использовать внешние скрипты для поиска данных по другим информационным ресурсам.



› Поиск по Amazon пока приостановлен, но команда разработчиков *Tellico* сулит вернуть его в версии 2.x.

Помимо обычного поиска, на основе любых полей в коллекции создаются фильтры. Сохранив эти фильтры, можно применять их для вывода выборок из коллекции несколькими щелчками мыши.



Вердикт

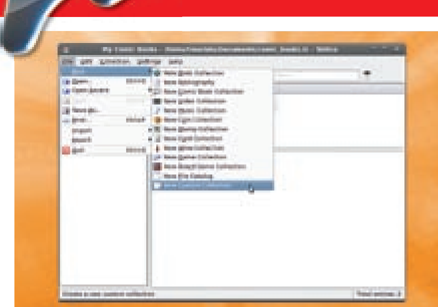
Tellico

Версия: 1.3.5
Сайт: www.periapsis.org/tellico
Цена: Бесплатно под GPL

» Дизайн позволяет легко добавлять данные, а шаблоны можно редактировать; но интерфейс мог бы быть и краше.

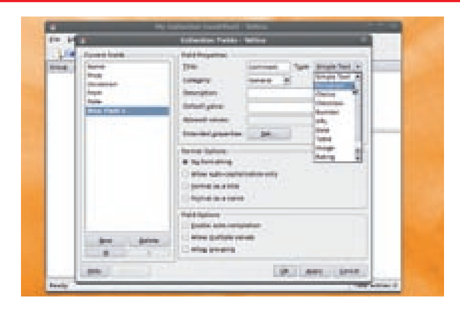
Рейтинг 8/10

Шаг за шагом: Создание коллекции своего типа



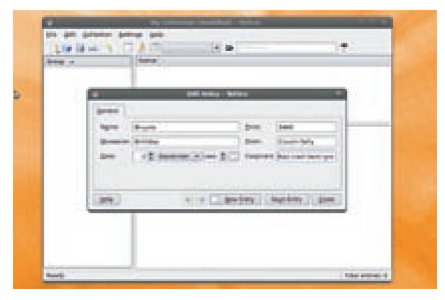
» Создайте коллекцию

Нажмите Файл > Новый > Личная коллекция для создания нового шаблона, затем выбирайте поля для этой коллекции. Не пытайтесь сразу довести все до идеала: поля можно добавить и позже.



» Добавьте полей

Нажмите Коллекция > Поля коллекции. Новые поля добавляются нажатием Новый. Нажмите Применить для сохранения значений каждого поля. Учтите, что свойства заполненного поля менять нельзя.



» Заполните данные

Для добавления данных, нажмите на Коллекция > Новая запись. Поле Комментарий имеет свою вкладку и отличается размером от других полей. Нажимайте Сохранить запись после ввода данных и сохраните еще раз по завершении.

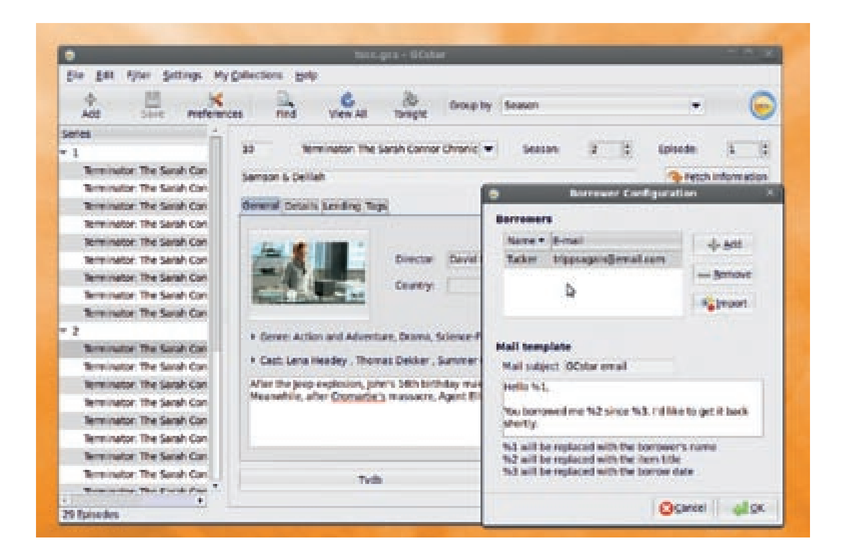
GCStar

Эта бета обещает больше, чем иное готовое приложение.

С выходом бета-релиза *GCStar* 1.5.0, версию приложения, хранящуюся в репозиториях большинства дистрибутивов, пора гнать на пенсию. Но данная «бета» не просто превосходит предшественников: она на голову выше также и своих противников.

Как и *Tellico*, *GCStar* поставляется со встроенными шаблонами для различных типов коллекций. Шаблоны было подзагнили, но бета внесла свежую струю, добавив возможность составления коллекции ТВ-сериалов и моделей автомашинок. И стоит вам запустить *GCStar*, как в глаза ударит внимание к деталям, красной нитью пронизывающее все шаблоны.

› Можно установить систему для отслеживания вещей из коллекции, которые вы одолжили друзьям или родным.



Внимание к деталям

Поскольку сайтов, предлагающих разработанный общими усилиями список моделей автомобилей, пока нет, придется вводить названия самим. Это, конечно, печально, но с шаблоном жизнь все равно проще, если у вас припасено много моделей.

Для другой своей новинки, *GCStar* умеет подключаться к сайту TVDB, чтобы извлекать информацию о телесериалах. Сведения на этом сайте предоставляются пользователями, и есть риск, что они неполны или неточны; не исключено, что кое-что вам придется вводить вручную.

Допускается контроль над тем, что именно отображать по каждому элементу вашей коллекции: не хотите, скажем, включать поле Комментарии [Comments] или В ролях [Cast] – ну так и удалите их. Изменение состава полей влечет изменение количества диалогов ввода, и у вас не запросят лишней информации.

В *GCStar* встроен список сайтов, откуда можно получить данные для каждого типа коллекции, и вы можете выудить добавочные сведения из активных форумов сообщества. Можно также настроить, при каждом ли добавлении нового пункта запрашивать источник, и даже установить сайт-источник отдельно для каждого поля.

Система по займам в *GCStar* также достойная: она предусматривает отправку писем заемщикам, когда одолженную вещь пора вернуть.

Одним из крупнейших барьеров к использованию *GCStar*, однако, является отсутствие поддержки печати. Это означает, что для распечатки отчета придется экспортировать данные в HTML, а затем прибегнуть к помощи браузера; альтернативно, имеется экспорт сведений в XML или в tar-архив.



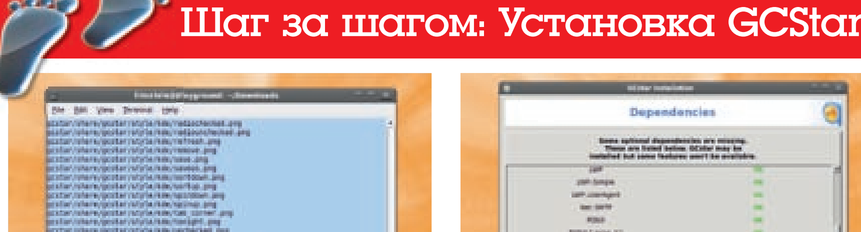
GCStar

Версия: 1.5.0.beta1
Сайт: www.gcstar.org
Цена: Бесплатно под GPL

» Приложение великолепно – новые шаблоны, удобный интерфейс и приличная документация, в одном пакете.

Рейтинг 9/10

Шаг за шагом: Установка GCStar

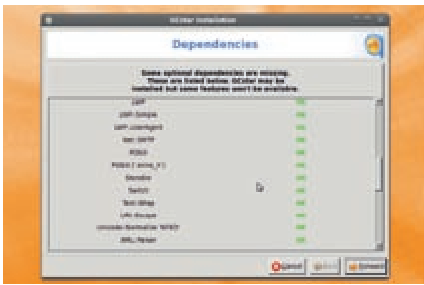


» Распакуйте tar-архив

Откройте окно терминала, найдите директорию, содержащую **gcstar-1.5.0.beta1.tar.gz**, и введите

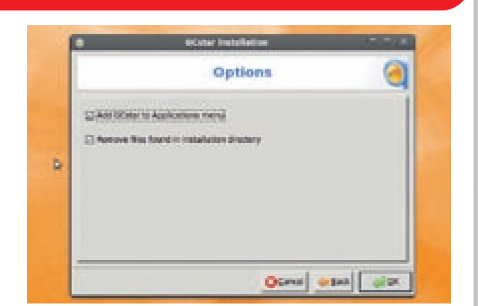
```
tar zxvf gcstar-1.5.0.beta1.tar.gz.
```

Перейдите во вновь созданный каталог **gcstar/** и введите **sudo ./install**.



» Сценарий установки

Скрипт запустит графический инсталлятор. Первый экран проинформирует вас обо всех зависимостях, установленных на вашей системе, требуемых для *GCStar*. При нехватке зависимостей продолжить вы не сможете.



» Добавляем в Приложения

Нажмите Вперед и выберите место установки *GCStar*. По умолчанию это **/usr/local/bin**. Теперь отметьте Добавить GCStar в меню приложений, а потом запускайте его через Приложения > Офис > Менеджер коллекций GCStar.

Data Crow

Программа делает все, но не ценой ли излишних жертв?

Впервые выпущенный в 2003 году, *Data Crow* за несколько лет вырос в одно из самых популярных программных средств каталогизации. Для приложения под GPL на основе Java, здесь предлагаются функции, совершенно невиданные в аналогичных инструментах. Можно генерировать отчеты и получать данные из Интернета для ускорения создания каталога коллекции; а множество удобных мастеров помогают выполнить почти все, будь то добавление новых элементов или редактирование встроенных модулей. Мало того, можно создать диаграмму, представляющую вашу коллекцию графически, и гармонизировать ее внешний вид с вашим рабочим столом через прилагаемые темы.

Также предлагаются обширные опции поиска и фильтрации и умный интерфейс. Но, пожалуй, тут не без горя от ума. Стремясь превзойти всех в каталогизации данных, *Data Crow* превратился в довольно сложную зверюгу.

Зато он не требует установки. После извлечения файлов из **datacrow_3_4_12_zipped.zip** вам останется только скомандовать

```
java -jar datacrow.jar
```

При первом запуске *Data Crow* вам предложат указать, Новичок [BegInner] вы или Пользователь-эксперт [Expert]. Выбор не роковой: вы в любой момент можете сменить свой статус через меню Опыт [Experience]. Разница будет в том, что новичкам нельзя создавать новые модули или изменять встроенные, такие как Книги [Books], Музыка [Music], Кино [Films] и прочее. Как ни странно, нет встроенного модуля для комиксов, хотя он основной во многих менеджерах коллекций.

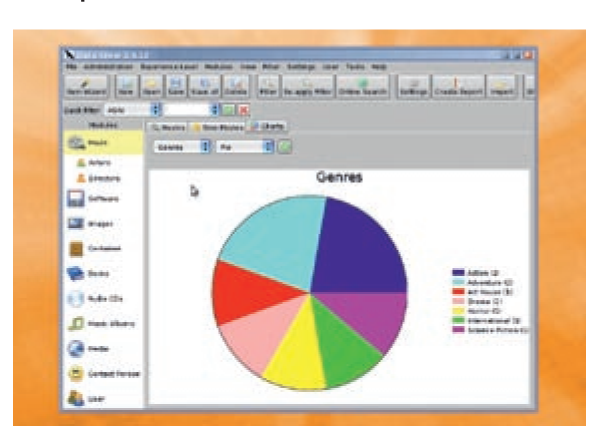
Интернет-магия

Теперь о создании вашей коллекции. Заручитесь для этого поддержкой одного из бригады умелых мастеров *Data Crow*. Просто введите ключевые слова или, для книг, номер ISBN – и *Data Crow* пошарит онлайн в поиске нужных совпадений. Только ничего не найдя, он предложит ввести данные вручную. *Data Crow* умеет извлекать сведения и прямо с музыкального диска, сильно упрощая каталогизацию фонотеки.

> «Tellico поддерживает массу интернет-источников.»

Приятно, что он может также создавать диаграммы для каждой из ваших коллекций на основе одного из полей данных и отслеживать вещи, временно одолженные друзьям, для чего нужно ввести данные вашего друга в коллекцию Контактных лиц [Contact Person]. Но вообще-то функция управления займам отслеживает, что и кому вы одолжили.

Наша основная претензия к программе – на сайте проекта нет документации, и к *Data Crow* приложена только не особо развитая встроенная справка, вызываемая нажатием Справка > Справка [Help > Help]. Полезных всплывающих подсказок очень не хватало во время нашего общения с *Data Crow*.



› **Диаграммы дают хорошее представление о вашей коллекции. Ну прямо цветик-семицветик.**



Вердикт

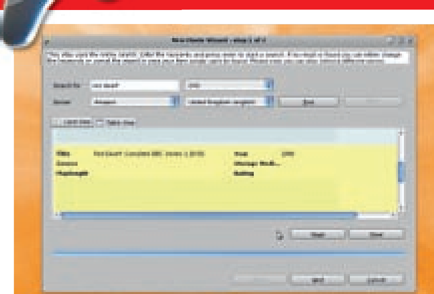
Data Crow

Версия: 3.4.12
Сайт: http://datacrow.net
Цена: Бесплатно под GPL

» Пытается взять на себя слишком много, и в результате интерфейс путаный; а лучше бы он включал всплывающие подсказки.

Рейтинг 8/10

Шаг за шагом: Работаем в Data Crow



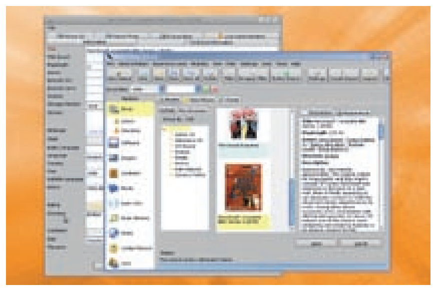
» Поиск в Интернет

Чтобы найти данные онлайн, введите ключевые слова, следя за полем Сервер, которое изменяется в зависимости от типа коллекции. Затем выберите подходящий элемент из списка результатов. Нажмите Далее внизу по завершении.



» Проверка информации

Data Crow примется собирать информацию, прикрепленную к выбранному элементу. По окончании, проверьте данные и внесите изменения, если что-то пропущено или неправильно. По завершении нажмите Закончить.



» Добавление в коллекцию

Наконец, нажмите кнопку Сохранить в панели инструментов для внесения этого элемента в вашу коллекцию. Потом вы сможете редактировать данные элемента двойным нажатием на названии или щелчком по иконке в средней панели; внеся изменения, не забывайте нажимать Сохранить.

Stuffkeeper

Разносторонний, полный функций и нюансов, но отнимает много времени.

Задуманный настолько универсальным, чтобы каталогизировать почти все на свете, *StuffKeeper* достигает этой цели не без апломба. И это впечатляет, тем более что приложение еще ожидает своего первого стабильного выпуска. Однако это программа из тех, которых одни любят, а другие ненавидят.

В отличие от большинства менеджеров, имеющих встроенные готовые шаблоны с кучей предопределенных полей, *StuffKeeper* предлагает создать их самим. Не спешите кривиться: зато в итоге вы получите аккуратную, не зияющую пустотами коллекцию. Плюс, ускоряется ввод элементов, потому что вы лучше представляете, какие данные нужны.

Для каждой коллекции можно создать сколько угодно полей, но учтите, что интерфейс достаточно базовый и требует времени на привыкание. И хотя на сайте есть беглый тур по функциям *StuffKeeper*, отсутствие подробной документации делает эту программу идеальной разве что для энтузиастов.

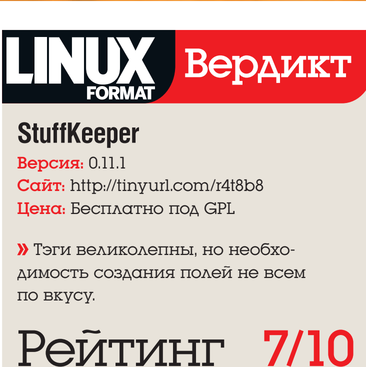
Чтобы помочь вам продраться через ваши введенные данные, можно создавать тэги для элементов различных коллекций и выполнять поиск по этим тэгам, просто набрав их в строке поиска. Кроме того, можно искать данные, введенные в любом поле вашей коллекции; а что самое главное, при этом не нужна точность. Например, вы можете найти все фильмы Лиам Нисон, набрав в строке поиска «Лиам»: искать ключевое слово будут в текстовом поле и во всех связанных с ним тэгах.

Наконец, стоит отметить, что *StuffKeeper* создает базу данных для каждой из коллекций и допускает резервное копирование данных в сжатые архивы.

> «Программа из тех, которых одни любят, а другие ненавидят.»



› Если вам нужен полный контроль над хранимыми данными, ваш выбор — *StuffKeeper*.



LINUX FORMAT Вердикт

StuffKeeper

Версия: 0.11.1
Сайт: http://tinyurl.com/r4t8b8
Цена: Бесплатно под GPL

» Тэги великолепны, но необходимость создания полей не всем по вкусу.

Рейтинг 7/10

Moll

А не лучше ли взять простой текстовый редактор?

Возможно, единственный инструмент командной строки среди программ данного назначения, *My Collection Manager* (известный как *Moll*), едва ли достаточно полезен, чтобы вызвать на бой продвинутые графические альтернативы из нашего списка.

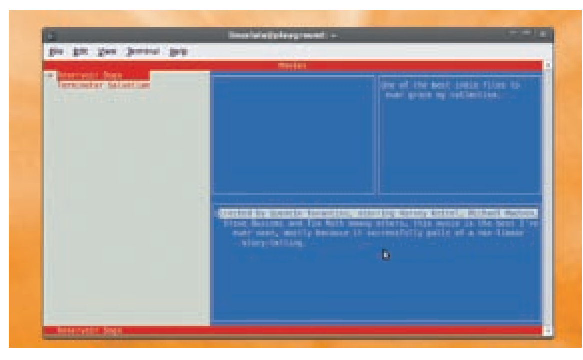
По сути, все, что он предлагает – большие пустые текстовые поля, которые можно заполнить данными об элементе. Интерфейс довольно скудный, но программа поставляется с встроенной документацией, получаемой по нажатию клавиши F1. Увы, при использовании *Gnome Terminal* для доступа к *Moll* нажатие F1 вызовет, вместо конкретной справки по программе, помощь терминала, что крайне неудобно.

Создание коллекции – столь же неблагодарное дело. Настройка только одна: вы можете ввести некоторый текст описания, который отображается в одной из многочисленных панелей интерфейса выбора коллекций. Для каждого элемента коллекции *Moll* также позволяет добавить краткое описание. Количество слов, отведенное на это, ограничено, поэтому будьте проще. Все прочее, что вы хотите зафиксировать об объекте – например, автор, название, издатель, исполнители и так далее – оставляется на большее поле Описание.

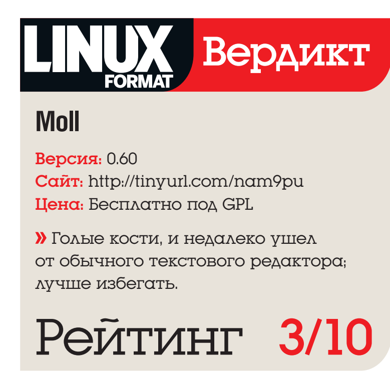
Вот и все функции *Moll*. Нет поиска, нет возможности фильтровать коллекции, нельзя сортировать или экспортировать введенные данные.

Честно говоря, для хранения ваших коллекций программа подходит немногим лучше, чем простой текстовый редактор, и единственное преимущество ее использования – то, что можно создавать различные коллекции с любым количеством элементов, и для хранения сведений об элементах коллекции предусмотрена связь с базой данных Беркли.

> «Интерфейс скудный, но программа идет с документацией.»



› Интерфейс Moll напоминает нам о нашей любви к миру *Ncurses*.



LINUX FORMAT Вердикт

Moll

Версия: 0.60
Сайт: http://tinyurl.com/nam9pu
Цена: Бесплатно под GPL

» Голые кости, и недалеко ушел от обычного текстового редактора; лучше избегать.

Рейтинг 3/10

Менеджеры коллекций

Вердикт

GCStar 9/10

Когда речь заходит о мире менеджеров коллекций, ясно, что вы не должны ограничиваться новаторскими решениями вчерашнего дня. Эти скромные каталогизаторы заметно улучшаются с каждым новым релизом, и, вырастая, поднимают уместный вопрос: какое участие вы хотите принимать в сборе данных о своей коллекции? Позиция по умолчанию – сделать этот процесс быстрым и безболезненным, с поиском сведений в Интернете и готовыми шаблонами, но для желающих принимать более активное участие существует немалый выбор вариантов.

Мы считаем, что брать нужно лучшее из обоих миров, предпочтительно с большим количеством готовых структур для обработки всего подряд, не упуская и возможность что-то приправить собственными усилиями. По этой части, *Moll* – наименее впечатляющий менеджер коллекций. Он не умеет хранить ничего больше, чем базовая информация, не поддерживает поиск или метки и не дает никаких способов экспорта ваших данных. Тем не менее он заслуживает похвалы за хорошую документацию и простой, но полезный интерфейс для просмотра вашей коллекции.

В то же время, *StuffKeeper* чересчур экстремален в своих возможностях настройки, и больше всего подходит любителям контролировать каждый аспект управления коллекцией. Если у вас есть время и склонность вникнуть в него, вы обнаружите немало отличных функций, поэтому не проходите мимо.

В итоге остаются три сходных программы-кандидата на первенство: *Data Crow*, *GCStar* и *Tellico*.

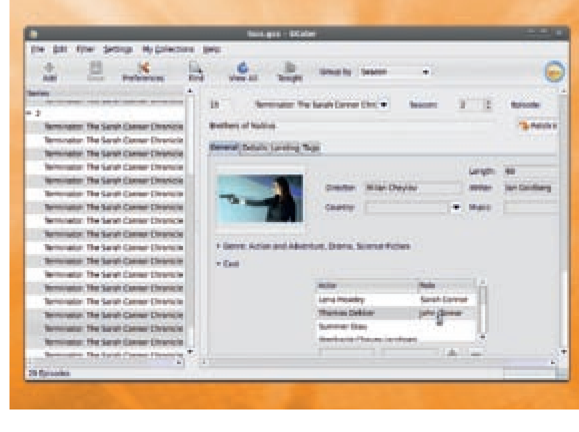
Главарь шайки

На первый взгляд, *Tellico* казался очевидным победителем. Здесь есть встроенные шаблоны, настраиваемость и хорошая документация, дизайн элегантен и даже мил; но его обошла программа, повысившая планку доступного менеджеру коллекций.

Data Crow остается на третьем месте из-за своего угнетающего интерфейса. В него втиснуто слишком много полей и нет приличной справки, помогающей осилить море опций. Тем не менее, приложение делает все грамотно, имеет ряд прекрасных функций, которые вы больше нигде не встретите, и оно больше всех насыщено темами; но уж слишком стремится урвать все, и это стоило ему победы.

> «GCStar произвел на нас впечатление изобилием модулей.»

Наконец, есть *GCStar*, который произвел на нас впечатление изобилием модулей для подключения ко всяческим источникам данных. Кроме того, у него самый внятный интерфейс, особенно по части редактирования элементов вашей коллекции. Большинство инструментов требуют дважды нажать на элементе, который необходимо изменить, а затем перейти на поля в другом диалоговом окне, а *GCStar* позволяет редактировать поля без лишней суеты. Сайт *GCStar* очень оживленный, с активными участниками форума, и, приняв в расчет количество разработчиков и помощников, вы получаете привлекательное и активно развивающееся приложение. LXF



› GCStar стал звездой благодаря прекрасному интерфейсу и реально полезным функциям, потрафив даже опытным пользователям.

Обратная связь

Думаете, победы заслуживал *Data Crow*, за его уникальные функции? Влияет ли на ваш выбор интерфейс – будь то *GTK*, *Qt* или Java? Присылайте ваши мнения на letters@linuxformat.ru

Таблица характеристик

Название	Лицензия	Интерфейс	Документация	Тэги	Ресурсы Интернета	Специфические коллекции	Редактирование полей	Фильмы	Игры	Музыка	Комиксы	Кроссплатформенность	Генерация отчетов	Управление займами
Data Crow	GPL	Java	✔	✔	✔	✔	✔	✔	✔	✔	✘	✔	✔	✔
GCStar	GPL	GTK	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Moll	GPL	CLI	✔	✘	✘	✔	нет[1]	✔	✔	✔	✔	✘	✘	✘
StuffKeeper	GPL	GTK	✔	✔	✘	✔	нет[1]	✔	✔	✔	✔	✘	✘	✘
Tellico	GPL	Qt	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

[1] Нет встроенных шаблонов и полей, поэтому нечего редактировать. Но и без встроенных шаблонов эта программа справляется со всеми типами коллекций.



Linux-ремикс

Оторваться от толпы! **Боб Мосс** покажет, как перекроить Linux по вашим лекалам.

Разработчикам основных дистрибутивов Linux приходится принимать важные решения, которые влияют на тысячи потенциальных пользователей. Нужно ли включать/исключать тот или иной пакет? Стоит ли приложить заплатку, нарушающую совместимость со старыми компьютерами? Эти вопросы яростно обсуждаются на форумах, где пылает пламя, рычат тролли, а лидеры проектов отбивают от своих решений атаки несогласных.

Но лично у вас подобных проблем нет. Вы можете устанавливать и удалять какие угодно пакеты, и сами определять, ставить несвободное ПО на вашу систему или нет. И зачем вам идти на поводу у сообщества,когда можно направить Linux по своему пути?

Дистрибутив, который вы установили, скорее всего, содержит драйверы и компоненты, которые вам не нужны. Мало того, программы, которые вы добавляете на каждой своей системе, будут скачиваться и устанавливаться всякий раз заново, что неудобно и отнимает уйму времени.

> «Мы охватим области, где можно подогнать систему под себя.»

Мы подошли к основной мысли. Если вы уделяете программированию больше времени, чем играм, то можете заменить развлекательные пакеты на вашу любимую IDE. А почему бы не удалить драйверы устройств, которые вам в жизни не понадобятся, сэкономив место на диске? Те из вас, кто последовал советам из статьи «Ускорим Linux» в прошлом месяце, могут даже применить обсуждаемые там подстройки автоматически на стадии установки.

Собрались ли вы подогнать систему под конкретную машину или затеваете сбросить Ubuntu с пьедестала почета, найдется множество инструментов вам на подмогу. Одни просты в использовании, но имеют меньший потенциал; другие сложнее в настройке, зато дают вам максимальный контроль. Мы рассмотрим подводные камни и охватим области, где можно подогнать систему под себя, но в конечном итоге все права за вами – внедряйте столько изменений, сколько захотите: ведь это ваш Linux.



Ubuntu Customisation Kit

Подстройте Ubuntu под себя этой простой утилитой.

Если вы уже сыты по горло установкой одних и те же пакетов при каждом развертывании Ubuntu, вы найдете, что проще всего будет сделать так, чтобы ваш установочный диск уже включал эти пакеты. Вам потребуется 5 ГБ свободного места на винчестере, ISO-образ установочного диска Ubuntu и *Ubuntu Customisation Kit*, установленный через менеджер пакетов. Обзаведясь этими ресурсами, просто запустите *UCK* и следуйте экранным инструкциям. Вы увидите, что львиная доля обычных шагов, которые вам приходилось делать в прошлом при создании Live CD, теперь автоматизирована. Добавьте к этому интуитивность GUI, и легко заметить, что при желании каждый сможет без трудов и быстро создать свою копию Ubuntu. Можно отредактировать Ubuntu насколько вам угодно (см врезку), но тогда уж примите весь огонь на себя.

Типичная ошибка при работе с *UCK* – пропуск диалога, позволяющего выбрать, какие пакеты ставятся по умолчанию, а какие нет. В итоге получается стандартный установочный диск Ubuntu, только с другим именем, а от этого проку мало! Чтобы гарантировать себе доступ к *Synaptic*, ответьте «Yes» на вопрос, хотите ли вы настраивать CD вручную во время сборки. При этом выведется диалог, предлагающий выбрать *Synaptic* или терминал, либо продолжить установку.

Решения, решения

В нашем примере (креативно названном BobBuntu) мы добавим репозиторий Medibuntu (см. руководство по PPA Ubuntu в **LXF124**). Для этого из диалога откроем терминал и выполним такую команду:

```
wget http://www.medibuntu.org/sources.list.d/jaunty.list
--output-document=/etc/apt/sources.list.d/medibuntu.list
```

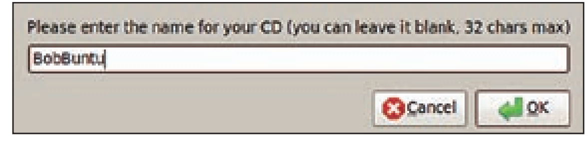
Затем можно установить GPG-ключ для этого репозитория, введя

```
apt-get update && apt-get install medibuntu-keyring && apt-get
update
```

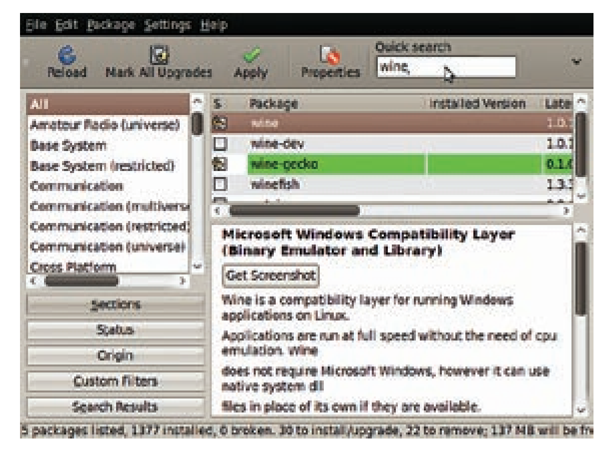
Этот репозиторий третьей стороны дает доступ во всем разновидностям несвободного ПО, как то: *Adobe Reader*, *Skype* и кодеки проприетарных форматов мультимедиа. Теперь, запустив из диалога *Synaptic*, можно увидеть, что пакеты *Skype* доступны при поиске (может быть, для этого придется обновить список пакетов).

Мы уберем из BobBuntu *OpenOffice.org*, заменив его на *AbiWord* и *Gnumeric*, затем удалим *Ekiga* и откажемся от *Firefox* в пользу *Epiphany*. Хотите – включите сюда понравившиеся приложения из Launchpad PPA и любого другого репозитория третьей стороны, или уберите редко используемые вами приложения, просто ставя и снимая птички у соответствующих пунктов.

Применив изменения щелчком по Apply, вы перенесетесь на стадию сборки, при этом в каталоге **~/tmp/remaster-newfiles/** создастся образ Live CD под именем **lived.iso**. По завершению сборки из этого образа можно будет создать CD, или испытать образ в пакете вроде *VirtualBox*, чтобы потом испробовать свой личный respin Ubuntu.



› **На стадии настройки вы можете дать своему диску образное (или чисто эгоцентрическое) имя.**





› **В *Synaptic* вы можете своей панской волей решать, какие пакеты включать, а какие — нет.**

» Сложность
Легко

» Нужное время
30 минут

» Перенастройка
Малая

У *Ubuntu Customisation Kit* замечательный набор возможностей. Например, можно легко выбирать пакеты, рабочие столы или локальные настройки, или включить туда *Wubi*, если вы решили распространять свой респин среди пользователей Windows.

Однако в *UCK* явно недостает функций, что заставляет обратиться к другим сервисам. Так, нельзя задать внешний вид своего респина Ubuntu без усиленного терзания командной строки или создать загрузочный образ для Live USB (хотя это можно сделать из вашего нового Live-окружения). И вы прикованы к Ubuntu как к родительской системе: конечным продуктом будет все-таки вариант Ubuntu. Тем не менее, *UCK* – отличное введение в разработку дистрибутивов и идеальное средство, если вам просто нужна копия Ubuntu с уже готовым любимым рабочим окружением.

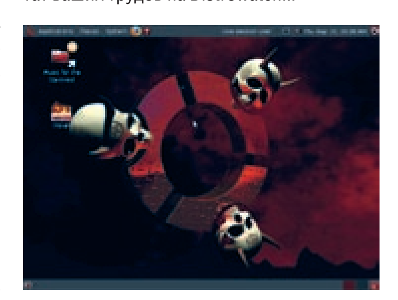
Варианты Ubuntu

Ubuntu славится простотой настройки, и поэтому встречается до смешного огромное число вариантов, рассеянных по Интернету, типа nUbuntu, Xubuntu… есть даже версия для сатанистов.

Некоторые из этих вариантов содержат расширения, обогащающие опыт пользователя. Например, Ultimate Edition снабжен такими программами, как *GDM Maker* и *USplash Maker*, чтобы пользователи могли создавать свои темы для экрана входа в систему и для заставки; отдельная версия предназначена для геймеров, со множеством предустановленных игр. Темы и выбор устанавливаемых приложений (свободных и несвободных) рассматриваются под углом постройки «наилучшего дистрибутива».

Есть и другая крайность. Например, Mundus OS комплектуется с *Alien*, чтобы можно было устанавливать не-Ubuntu-вские пакеты, и пресловутым решением для монтирования файлов **.dmg**, а в остальном это просто обычный ISO Ubuntu, который можно скачать с сайта сообщества.

Количество изменений определяется только вами, будь то совершенно новый способ распространять пакеты или просто веселенькие обои. И кто знает: если людям понравится то, что вы сделали, то в недалеком будущем вы сможете увидеть результат ваших трудов на Distrowatch...



› **Респин Ubuntu найдется для любой ниши.**



SUSE Studio

Создавайте собственные варианты SUSE прямо из браузера.



» **Сложность** Легко
» **Нужное время** 30 минут
» **Перенастройка** Средняя

Недавно Novell выпустил SUSE Studio, службу, позволяющую создавать респины SUSE из любого браузера. На момент подготовки номера эта служба была сильно ограничена и действовала только по приглашениям, хотя его можно было запросить на www.susestudio.com.

Даже в бета-версии она оказалось мощным и легким в использовании приложением. Первичная цель данной службы – позволить вам генерировать виртуальные программные устройства [appliance] с SUSE в качестве базы, почти для любой цели. А наша цель – создать с ее помощью собственный респин дистрибутива. Потенциал для подстройки дистрибутива тут огромный. У вас есть выбор, какую версию SUSE использовать, а затем указывается базовый пакет, который определит, какой рабочий стол и платформу вы, скорее всего, примете для своего респина. В качестве рабочего стола можно взять Gnome, KDE или LXDE, или вообще решить работать на сервере. Мы рекомендуем остановиться в выборе платформы на OpenSUSE, это больше подойдет для домашних пользователей; однако при желании можно взять SUSE Enterprise.

> «SUSE Studio дает вам много больше свободы в создании респина.»

Следующие опции позволяют указать, какие программы будут включены или исключены, используя средство поиска внизу страницы. Щелкнув по соответствующей ссылке, можно добавлять другие репозитории и пользовательские RPM'ы. В каждом случае интерфейс выглядит логичным для службы, основанной на браузере, и это на удивление интуитивно.



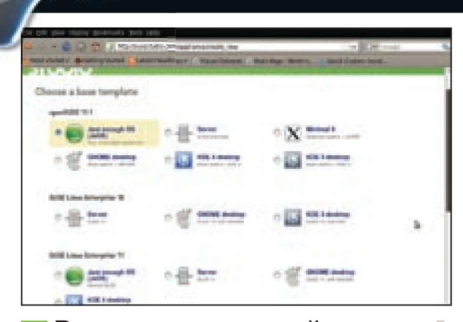
› **Следуя нашему пошаговому руководству, вы сможете сделать и свое хваленое виртуальное устройство SUSE. Конечно, о вкусах не спорят!**

Личное хранилище файлов

Затем можете указать внешний вид, выбрав предоставленные логотипы и обои; можно загрузить и собственные, чтобы придать респину налет индивидуальности. Выгрузка необходимых файлов в каталог **/home** вашего респина – также процедура безболезненная, это бонус для тех, кого страшит самая мысль о переносе своей драгоценной коллекции музыки при каждой новой установке.

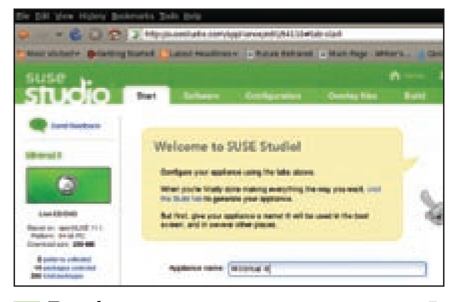
Параметры настройки весьма обширны: они позволяют определить локаль, разрядность (32 или 64 бита), любые приложения или службы для запуска при старте, и даже виртуальный жесткий диск, если вы хотите создать виртуальное устройство, а не Live CD. Гибкость и потенциал этого сервиса превосходны, и это один из простейших способов создания респина SUSE, заточенного под ваши конкретные нужды. И при всей доступности параметров настройки, SUSE Studio устанавливает для виртуального устройства разумный набор умолчаний, зависящий от выбранного базового пакета, так что можно просто вносить изменения где хотите, не беспокоясь о формальностях. Использовать SUSE

Шаг за шагом: Создаем устройство в SUSE Studio



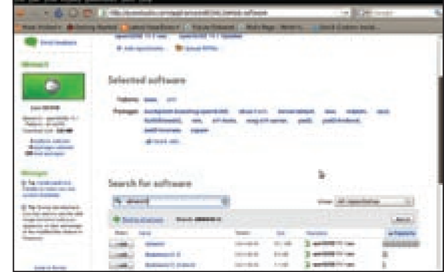
1 Виртуальное устройство

Здесь выбираются рабочий стол, платформа и версия SUSE, простым щелчком на переключателе.



2 Выберите имя

Текущая статистика о вашей машине доступна слева, и обновляется при каждом изменении.



3 Программы

Здесь можно поискать пакеты, чтобы добавить или убрать их, или указать дополнительные репозитории, или загрузить собственные RPM'ы, щелкая по соответствующим кнопкам.

Studio не сложнее, чем любой другой сайт, отвечающий стандартам, а навигация позволяет пользователям запросто перемещаться вперед и назад через этапы создания, просто щелкая по нужным вкладкам.

Прекрасная служба?

По сравнению с *Ubuntu Customisation Kit*, SUSE Studio дает много больше свободы в создании собственного респина, в особенности потому, что вы можете иметь доступ к службе с любого хоста, где есть браузер. Однако при не слишком быстром интернет-соединении скачивание вашего конечного виртуального устройства или Live CD станет серьезным препятствием, особенно обидным оттого, что существуют и оффлайн-утилиты для тех же задач.

Даже при достаточно быстром для этого сервиса интернет-подключении, вас может оттолкнуть то, что создаются только респины SUSE – хотя чего еще и ожидать от сервиса, созданного Novell (который владеет и поддерживает SUSE). Однако по окончании бета-стадии код SUSE Studio будет опубликован под открытой лицензией, и скоро должны появиться подобные сервисы и для конкурирующих дистрибутивов. SUSE заточен под нужды предприятия и хорошо поддерживается сообществом Linux, но если вам до смерти охота сделать респин другого дистрибутива, хорошей альтернативой будет Instalinux (см. врезку ниже).

Другие инструменты настройки

Revisor

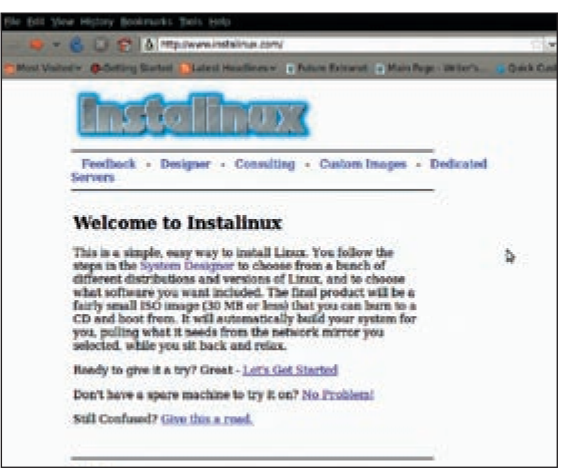
Если вам понравился вид *Ubuntu Customisation Kit*, но вы предпочитаете Fedora, то вам повезло. Просто скачайте *Revisor*, используя ваш менеджер пакетов, и сразу приступайте к созданию респина. *Revisor* не требует для своей работы исходного ISO, так как пакет автоматически создает новый ISO-образ, который можно записать на диск, или загрузочный образ, который можно перенести на Live USB; хотя тут не обойтись без хорошего интернет-соединения, чтобы скачивать основные пакеты.



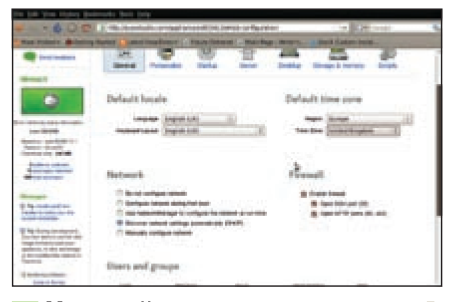
› *Revisor* доступен в репозиториях Fedora, а работает почти так же, как *UCK*.

Instalinux

Если вам нравится идея SUSE Studio, но нет сил дожидаться приглашения, обратитесь на www.instalinux.com. Интерфейс не столь отполирован, как у SUSE Studio; вам, вероятно, будет не хватать возможности изменять дизайн по своему вкусу и создавать виртуальные программные устройства. Зато, в отличие от SUSE Studio или *Ubuntu Customisation Kit*, здесь можно основывать свой респин почти на любом дистрибутиве, и сравнение показывает, что размеры созданных образов получаются меньше.



› *Instalinux* — метод создания респина через браузер, не ограниченный одним дистрибутивом.



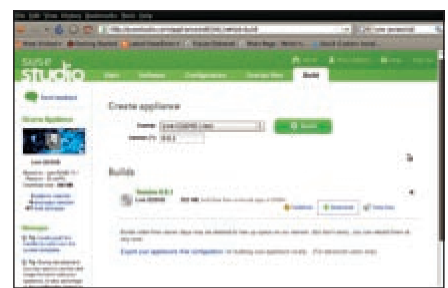
4 Настройка

Смело меняйте основы вашей системы, включая добавленных пользователей, локаль, стартовые приложения и любые скрипты.



5 Дизайн

В той же вкладке Configure можно выбрать логотип и обои, или загрузить собственные. Окно внизу экрана покажет вид вашего виртуального устройства.



6 Сборка

Мы выбрали сборку ISO-образа, но можно и подготовить виртуальное программное устройство для использования с *VMware Player* или образ для запуска с флэшки.



Arch Linux

Легкий и мощный дистрибутив, разработанный для создания респинов.



» **Сложность** Средняя
» **Нужное время** 90 минут
» **Перенастройка** Серьезная

Arch Linux был специально разработан так, чтобы пользователи могли настраивать его от самых основ. Вы скажете: «Ну и что? В вашей статье все такие!» Это правда, однако в отличие от проектов типа *UCK* и SUSE Studio, Arch Linux придерживается непрерывных релизов [rolling updates], чтобы исправлять ошибки и проблемы с безопасностью, так что вы можете посвятить больше времени на выбор рабочего стола или офисного пакета, чем на обдумывание порядка компиляции пакетов и их зависимостей. Это также означает, что у релизов Arch Linux нет номеров версий: каждая версия автоматически обновляется до следующей, так что он феноменально стабилен, а это важный фактор при выборе основы для вашего респина. Итак, базовый пакет Arch Linux – прекрасное начало, если нужно принимать более основополагающие решения, чем предлагаемые *Ubuntu Customisation Kit* или SUSE Studio.

Изначальная установка фундамента системы практически одинакова во всех случаях. Дается возможность задать, какие базовые пакеты и драйверы устанавливать, однако оставшаяся часть процесса касается лишь указания часового пояса и локали, а также установки ядра и загрузчика на раздел жесткого диска. Также от вас попросят вручную задать несколько конфигурационных файлов; но если это ваш первый опыт, можете спокойно брать предлагаемые по умолчанию в главе вики Arch Linux, на которую дана ссылка в пошаговом руководстве. Преимуществом использования вики является то, что позже всегда можно отредактировать эти файлы, чтобы приспособить Arch Linux к своим нуждам.

> «Конечный результат намного более гибкий и управляемый.»

Перезагрузившись после установки базовой системы (см *ниже*), нужно пройти еще несколько шагов, прежде чем заняться установкой рабочего стола, оконного и файлового менеджеров и прочего. Во-первых, введите в экране консоли **adduser** и следуйте инструкциям, чтобы добавить пользователя и пароль, под которым будет производиться вход в систему. Затем обновите систему, выполнив

```
pacman -Syu
```

Вы, поди, уже угадали, что **pacman** – это наш менеджер пакетов, но данная стадия не отнимет много времени, так как сейчас у нас имеется только минимальная установка.

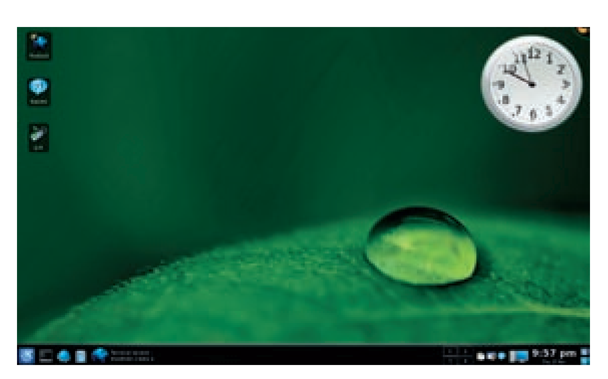
По умолчанию Arch Linux применяет *su*, но если вы предпочитаете использовать для получения полномочий root *sudo*, установите его и добавьте нашего пользователя в список sudo'шников, выполнив следующие команды:

```
pacman -S sudo
visudo
```

В появившемся файле сразу под **root(ALL)=(ALL)ALL** добавьте:

```
user(ALL)=(ALL)ALL
```

заменив **user** на имя пользователя, выбранное во время настройки.



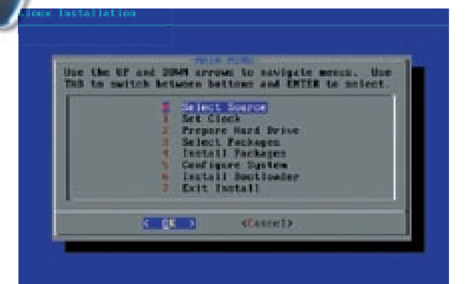
› **Arch Linux предлагает смесь стабильности и произвола выбора, идеальную для создателей дистрибутивов средней руки.**

Установка Xorg

Теперь можно начать устанавливать GUI, а наиболее важный его компонент – *X.org*. Для установки двоичного пакета введите:

```
sudo pacman -S xorg
```

Шаг за шагом: Установка базы Arch Linux



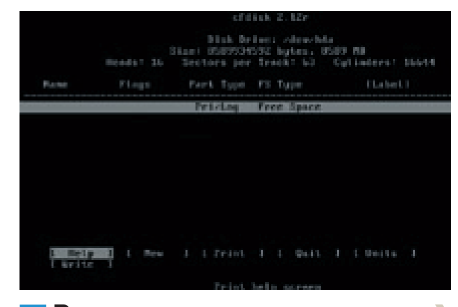
1 Главное меню

Добро пожаловать в мир текстовых установщиков! Для установки базовой системы Arch проделайте путь от пункта 0 до 7, следуя экранным инструкциям.



2 Локальные настройки

Определив свой метод установки, просто наберите правильные числа, чтобы указать свою страну и часовой пояс.



3 Разделы

Вы можете разбить свой жесткий диск с помощью подходящей GUI-утилиты, или использовать *cfdisk*, если любите делать все своими руками.

Если у вас графическая карта Nvidia, то для установки ее драйверов, настройки и применения к текущей инсталляции выполните

```
sudo pacman -S nvidia
sudo nvidia-xconfig
cp /etc/skel/.xinitrc ~
```

Пользователи графических карт ATI найдут более полную информацию на http://bit.ly/QLrGV, поскольку каждая карта склонна использовать свой драйвер и требовать дополнительной настройки. По завершении этого шага используйте последние две строки, заменив в них **nvidia-xconfig** на **ati-xconfig**.

Установка Gnome

Вы можете установить любой рабочий стол, какой ни взбредет вам в голову. Например, для добавления Gnome и всяких его дополнительных утилит выполните две такие команды:

```
pacman -S gnome
pacman -S gnome-extra
```

Оба раза вас спросят, хотите ли вы установить все сразу или выполнить тонкую настройку; но вы просто следуйте экранным инструкциям по установке рабочего стола Gnome.

Затем откройте файл **/etc/rc.conf**, отыщите в нем ту часть, где написано DAEMONS, и убедитесь, что она включает **portmap**, **fam** и **hal** где-нибудь в скобках. Если вы установили в качестве менеджера входа *GDM* (имя пакета – **gdm**), его можно настроить, открыв **/etc/inittab**, найдя следующие строки:

```
# Boot to console id:3:initdefault:
# Boot to X11 #id:5:initdefault:
```

и поменяв их на:

```
# Boot to console #id:3:initdefault:
# Boot to X11 id:5:initdefault:
```

Теперь пролистайте вниз и раскомментируйте следующую строку, убрав из нее знак решетки:

```
x:5:respawn:/usr/sbin/gdm -nodaemon
```

Сохраните файл и выйдите. Если вы хотите, чтобы Gnome запускал при старте приложения и скрипты, выполните следующую команду:

```
~/.config/autostart/*
```

Установка KDE

Для установки базового рабочего стола KDE 4 наберите

```
pacman -Sy kde
```

Если вам по сердцу идея сделать бальзам для глаз и поставить больше модулей расширения, установите также следующие пакеты:

```
pacman -S kde-extragear
pacman -Sy qtcurve-gtk2 qtcurve-kde4
```

Затем можно установить KDM, добавив пакеты

```
pacman -Sy kdebase kdebase-workspace
```

открыв **/etc/inittab** и поменяв файл тем же способом, что и для Gnome; но на сей раз замените **gdm** на **kdm**. Кроме того, обратите внимание на проект Chakra (www.chakra-project.org), собирающий для Arch альтернативные KDE-пакеты (включая версию 3.5) под маркой KDEmod.

Расширить и углубить

Теперь у вас есть базовая установка Arch Linux, которая автоматически обновляется, и вы можете добавлять в систему любые пакеты. Установите ли вы *OOo*, или предпочтете *AbiWord* и *Gnumeric*? Вы можете установить *Firefox* с вашими логотипами (об этом рассказано на http://bit.ly/T8ZE4). Другая интересная возможность – замена всего рабочего стола на *Openbox* и использование *Avant Window Navigator* для запуска приложений. Если вы захотите создать Live CD на основе ваших усилий, запустите *larch*, скрипт, генерирующий ISO из образа жесткого диска вашей установки. Его можно скачать с http://larch.berlios.de.

Arch Linux требует гораздо больше усилий, нежели *UCK* и SUSE Studio, и у вас нет полного контроля над организацией пакетов, однако конечный результат намного более гибкий и управляемый. Это лучшее, чего можно достичь при сборке собственного дистрибутива без компиляции каждого компонента из исходников...

Без паники!

Частая проблема, с которой сталкиваются на этапе сборки – пустой экран, без видимой реакции на действия клавиатуры и мыши. Хотя это и вселяет ужас, но данная проблема хорошо задокументирована и не является фатальной.

Одним из самых больших изменений в X-сервере, сделанных за последнее время, является отказ от фонового узора, который можно было наблюдать в дистрибутивах типа DSL перед загрузкой рабочего стола. Это можно исправить, установив белый фон:

```
X -wr -config xorg.conf.new
```

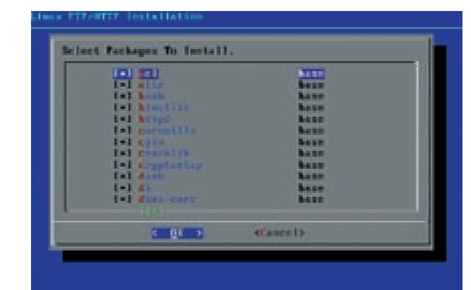
а если ваша клавиатура или мышь не определились, попробуйте добавить в **xorg.conf** следующие строки:

```
Section "ServerLayout"
  Option "AllowEmptyInput" "off"
EndSection
```

Вы увидите ряд строк, начинающихся символом #, между первой и второй строками – просто добавьте вторую строчку ближе к концу.

Другая досадная штука – недавнее удаление сочетания клавиш Ctrl+Alt+Backspace для перезапуска X-сервера, что в общем-то иногда приходится делать всем. Для его восстановления, добавьте такой кусок кода в конец файла **xorg.conf**:

```
Section "ServerFlags"
  Option "DontZap" "False"
EndSection
```



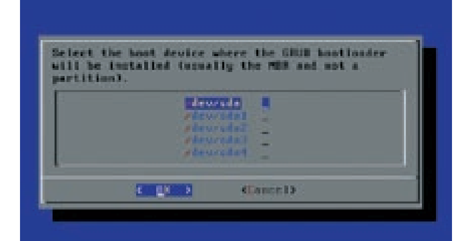
4 Пакеты

Задав требуемые категории, можно также выбирать пакеты индивидуально. Нажмите пробел для выбора/снятия выбора, а Enter для продолжения.



5 Настройка

Выберите ваш текстовый редактор (новичкам рекомендуем *Nano*). Примеры скриптов можно взять с вики Arch Linux на http://bit.ly/2vWyr.



6 Загрузка

При установке загрузчика укажите, с какого раздела вы хотите запускать *Grub*. Завершив установку базовой системы, приступайте к рабочему столу!



Linux From Scratch

Принимайте все решения с нуля, чтобы собрать дистрибутив-идеал.



» **Сложность**
Высокая
» **Нужное время**
210 минут
» **Перенастройка**
Наивысшая

Для истинных гуру Linux, кто любит все решать в дистрибутиве самим, от выбора драйверов ядра до определения, какой текстовый редактор использовать по умолчанию, *Vi* или *Emacs*, нет более дотошного пути, чем собрать свое ядро и каждую важную компоненту Linux с нуля. Здесь-то на сцену и выходит Linux From Scratch. LFS – это электронное руководство, созданное онлайн-сообществом с целью помочь заядлым рукодельникам построить собственный уникальный дистрибутив с наименьшей морокой.

Электронная книга (доступная на диске этого месяца) детально описывает все, что нужно знать, от сборки кросс-компилятора до использования *chroot* при настройке окружения пользователя. Однако есть несколько ошибок, про которые следует помнить, создавая свою LFS-систему. Естественно, мы приложили все усилия, чтобы у вас их не произошло, разработав уточнения от **LXF** к этой книге, и каждое из них заработано потом, кровью и бессонными ночами. Наш тест проходил на LFS 6.3, поскольку он сопровождался стабильным Live CD со всеми необходимыми нам утилитами, пакетами и заплатами. Однако если вы захотите испробовать последний писк LFS (на момент выхода номера это была версия 6.5), то все электронные книги рассматривают, как использовать для сборки LFS-системы ваш обычный дистрибутив. Какой путь вы ни выберете, наши советы должны помочь вам в беспроблемной установке!

> «LFS распахивает окно безграничных возможностей.»



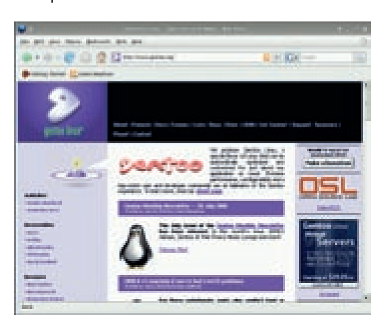
› **Электронная книга Linux From Scratch содержит четкое форматирование, и каждый шаг краток и по делу.**

Создание разделов

При первом запуске *cfdisk* распространенной ошибкой является выход без сохранения сделанных изменений (а потом еще удивляются, почему вдруг *mke2fs* возвращает ошибку «устройство /dev/hda1 не существует»), так что перед тем, как продолжить, дважды удостоверьтесь, что система имеет доступ к вашим разделам.

Попробуйте Gentoo

Если вы ищете дистрибутив с громадным потенциалом настроек, стоит взглянуть на Gentoo. Как и Arch Linux, Gentoo придерживается непрерывных релизов, но, в отличие от Arch, он позволяет выбирать, какое ядро использовать: Linux или BSD. Включено также содействие более 300 разработчиков, поддерживающих более 10000 пакетов, распространяющихся посредством того, что фанаты Gentoo называют самым мощным менеджером пакетов в мире Linux (имя ему *Portage*). Gentoo компилирует все пакеты из исходников и нацелен на стабильность, тогда как в Arch Linux устанавливаются двоичные пакеты, и поставленная цель – быть на переднем крае разработок ПО. То, что вы выберете, зависит от ваших целей и того, какую степень контроля вы предпочитаете, однако если вы разрываетесь между Arch и Linux From Scratch, Gentoo будет для вас неплохим решением.



› **Gentoo — мощная альтернатива для LFS и Arch Linux.**

Нахождение исходников

Если вы используете Live CD LFS, то все ваши утилиты, исходники и заплатки расположены в **/lfs-sources**. Рекомендуется скопировать их в **$LFS/sources**, предварительно создав пользователя lfs; это позволит затем производить большую часть работы, не обладая правами root. Нарвавшись на проблемы, убедитесь еще раз, что вашим рабочим каталогом является **$LFS/sources** и что команда

```
echo $LFS
```

возвращает **/mnt/lfs**.

Самонастройка

При первом запуске сборки *GCC* (с опцией «bootstrap») рекомендуем вам отойти попить чайку, а затем заварить еще одну чашку: этот этап занимает много времени, так как вы компилируете *GCC* три раза, а затем сравниваете третью сборку со второй. Смысл здесь в том, чтобы убедиться, что вторая и третья сборки одинаковые, а значит, *GCC* собран правильно уже в первый раз. Можно и пропустить самонастройку, но мы так делать не советуем: это чревато странными ошибками, которые впоследствии будет трудно распознать. Новые пользователи LFS, раскиданные по различным форумам, часто спрашивают, почему компиляция GCC происходит так долго и нормально ли это. Остальные в курсе, что это нормально, но ваше терпение будет вознаграждено гарантией отсутствия проблем в будущем.

Часто процесс сборки по команде **make** занимает столько времени, что вы успеваете забыть ввести **make install** после его окончания. Позаботьтесь, чтобы этого не случилось, так как потом у вас возникнут ошибки, и придется шагать назад и исправлять проблему.

Заголовки API ядра

Распространенная ошибка состоит в предположении, будто заголовочные файлы API ядра находятся в отдельном пакете. Это не так – вам придется извлечь пакет исходников ядра (обычно из **linux-2.6.x.tar.bz2**), а затем перейти в распакованный каталог, чтобы следовать инструкциям электронного руководства.

Выкрутасы командной строки

Прежде чем приступать к сборке LFS, важно изучить все тонкости *Bash*. Вы будете создавать символьные ссылки и писать многострочные команды (то есть там, где вы видите ****, нужно нажимать Enter). Однако вас может подловить форматирование электронной книги. По всему тексту многострочные команды не содержат > у каждой строки (хотя в терминале вы это увидите). А вот увидев > в электронной книге, постарайтесь ввести этот символ именно так, как написано. Мы забыли про это, вводя

```
gcc -dumpspecs | sed 's@^/lib/ld-linux.so.2@/tools&@g' \
> `dirname $(gcc -print-libgcc-file-name)`/specs
```

и получили странную ошибку о том, что файл не существует. Вторая строка в терминале должна была выглядеть так:

```
> > `dirname $(gcc -print-libgcc-file-name)`/specs
```

Нет — опечаткам

Если вы пользуетесь в качестве хост-системы обычным дистрибутивом, от души рекомендуем вам копировать с руководства команды оболочки. Главная тому причина – у вас будет меньше шансов написать что-нибудь неправильно. Опечатка может вызвать ошибку при дальнейшей работе (обычно это выливается в полную пересборку). Для тех, кто намерен набирать все самостоятельно или пользуется Live CD: семь раз проверяйте тип скобок и каждую вводимую команду, прежде чем запустить ее. Ручной ввод имеет смысл вначале, но в конце концов вам обязательно встретится монстр наподобие этого:

```
GCC_INCLUDEDIR=`dirname $(gcc -print-libgcc-file-name)`/
include &&
find ${GCC_INCLUDEDIR}/* -maxdepth 0 -xtype d -exec rm
-rvf '{}' \; &&
rm -vf `grep -l "DO NOT EDIT THIS FILE" ${GCC_
INCLUDEDIR}/*` &&
unset GCC_INCLUDEDIR
```

Мы попытались ввести это за один присест, но нечаянно использовали стандартные скобки – и угробили все проделанное ранее! Копированием этого можно избежать, или уж вводите каждую строку отдельно, разрывая команду по &&.

Будьте начеку

После второго прохода сборки *GCC* возникает соблазн делать компиляцию и установку пакетов просто командами

```
./configure --prefix=/tools
make
make install
```

Однако такое не может длиться вечно, и вскоре вы наткнетесь на пакеты, требующие обходных путей и заплат для правильной работы на вашей сборке LFS. По этой причине почаще сверяйтесь с электронной книгой, чтобы в дальнейшем избегать нештатного поведения и возможных ошибок.

Переход под root

Войти вы должны как пользователь lfs, но вскоре вам придется перебраться под root. Это можно сделать, удерживая клавишу Alt и правую стрелку (то же действие с левой стрелкой вернет вас к пользователю lfs). На этой стадии непременно сделайте резервную копию, как учит вас электронная книга. Команда **rm**, введенная в неправильное время и в неправильном месте, может свести на нет часы долгой работы. Можно создать архив (и, следовательно, сэкономить немного места), выполнив следующую команду:

```
tar -cvf --file=$LFS/tools/lfs_build.tar $LFS/tools
```

Затем скопируйте архив на внешний носитель или в такое место, откуда впоследствии будет легко восстановить систему. Архив вам пригодится если не как резервная копия, то для более быстрой сборки LFS в будущем.



› При наличии времени и усилий вы тоже можете создать нечто не хуже Nutyx.

Сборка дистрибутива-мечты

Linux From Scratch распахивает окно безграничных возможностей, и парная к нему электронная книга Beyond Linux From Scratch создана, чтобы расширить и сделать настраиваемым стандартный дистрибутив, основанный на LFS. Там обсуждается управление пакетами, а также сборка окружения рабочего стола и типовых пакетов.

Чтобы оценить охват Beyond Linux From Scratch, взгляните на Nutyx (www.nutyx.org). Это дистрибутив на французском, и на английский (и русский) перевода пока нет, зато в нем есть полноценный рабочий стол и набор приложений, то есть он вполне пригоден к использованию.

Ключевой момент в LFS состоит в том, что у вас есть полный контроль над тем, какие заплатки и обновления применять. Все компилируется из исходников, так что выполняется с быстротой молнии, и у вас всегда есть выбор, обновлять ли ядро и какие приложения оставлять в системе. Для достижения этой нирваны дистростроения требуются серьезные усилия и некоторое время, но результат того стоит. LXF

Обратная связь

Теперь у вас есть полный контроль над работой вашего дистрибутива, вы можете выполнять все что угодно. Сделаете ли вы дистрибутив для собственного употребления простым? Может быть, вы захотите завоевать свою нишу и построить вокруг него сообщество? Собрав дистрибутив, вы получите идеальный рабочий стол и узнаете, как на самом деле работает Linux. Мы будем рады узнать, как у вас идут дела, а если вы сделаете что-нибудь блестящее, мы попробуем отыскать для вашего творения место на диске одного из следующих номеров **LXF**.

Победа — в единстве

Глава Linux Foundation Джим Землин — о коде ядра, сотрудничестве и о том, почему Microsoft небезнадежна.



Мир Linux известен разбросом мнений. Мы можем заспорить на пустом месте, мы судим о людях по их оконным менеджерам, мы до смешного пристрастны к текстовым редакторам. Нам никак нельзя без некой организации, способной «популяризовать, защищать и стандартизировать» Linux. Последняя попытка сделать это — Linux Foundation; Джим Землин [Jim Zemlin] — ее исполнительный директор. Но: нужны ли нам кредитные карты, украшенные Туксом?

LXF: Этим летом Microsoft предприняла беспрецедентный шаг: предложила код к внесению в ядро Linux. Вероятно, вы были поражены...

Джим Землин: Думаю, они изрядно запоздали – другие компании сотрудничают с нами не один год. Мне кажется, это конструктивное движение – в том смысле, что оно опровергает всякие домыслы о непригодности GPL для распространения ПО. Еще несколько лет назад Microsoft ополчалась против GPL, а теперь принимает ее с распростертыми объятиями.

LXF: Вероятно, если бы ядро было лицензировано на условиях GPL 3 (сейчас действует GPL 2), то MS не стала бы предлагать свой код: в новой версии есть четкие статьи в отношении патентов.

ДжЗ: Об этом мы никогда не узнаем!



Особенный оптимизм Джиму внушает будущее открытого ПО в Восточной Азии.

LXF: До сих пор это была помощь, связанная с виртуализацией — код для упрощения запуска Linux на платформе Microsoft. Как бы то ни было, это ценный вклад в обеспечение функциональной совместимости.

ДжЗ: Совершенно верно.

LXF: Кто-нибудь из Linux Foundation вел переговоры с MS до того, как это произошло?

ДжЗ: Мы подобрались перед анонсом, но наше участие не потребовалось, ведь есть же сообщество разработчиков ядра. Хранитель подсистемы для данного направления – Грег Кроа-Хартман [Greg Kroah-Hartman], через него-то все и происходило. Мне кажется, подоплека всей этой истории в том, что на современном рынке ИТ невозможно выжить без сотрудничества. Microsoft опасается Linux как конкурента. Но еще больше Microsoft опасается *VMware*, вот и старается не отставать. С их стороны это вовсе не благотворительность: Microsoft преследует свои бизнес-интересы. В данном случае бизнес-интересы довели до участия в Linux-сообществе на условиях его модели коллективного развития, в форме передачи кода в ядро на условиях GPL 2.

> **О СОВМЕСТНЫХ ЗАБОТАХ**
> «На современном рынке не выжить без сотрудничества.»

LXF: Вы не могли не заметить, что двумя бриллиантовыми — высшего уровня — спонсорами OSCON 2009 были Microsoft и Intel. Как, по-вашему, это характеризует сообщество открытого ПО?



ДжЗ: Я думаю, глубоко копать здесь не стоит: на данном этапе эти компании посчитали нужным расширить связи с сообществом разработчиков, вот и все. Microsoft определенно наверстывает упущенное. Что до Intel, то когда они не были платиновыми спонсорами? Это повторяется ежегодно!

LXF: А теперь у Linux Foundation есть и своя конференция, LinuxCon.

ДжЗ: Мы каждый год проводим немало встреч. Весной на саммит по совместной работе мы смогли пригласить только членов нашей организации. У нас тогда было мало места, и пришлось ограничиться только ведущими специалистами от каждой области.

LinuxCon – это встреча, открытая для всех желающих. Нам хотелось бы вернуть уютную атмосферу давних Linux-конференций, дискуссии, обмен глубокими техническими идеями. На таких встречах люди, от которых зависело принятие решений, обсуждали бизнес-предложения о развитии мира Linux.

LXF: Планируете проводить событие ежегодно?

ДжЗ: Обязательно. Кроме того, мы проводим встречу в Японии: это первая Linux-конференция такого масштаба в Азии. Как вы, вероятно, знаете, в Азии я провожу немало времени, и рост влияния Linux в этом регионе просто поразителен: от изготовителей оборудования в Тайбэе до китайских, корейских и японских производителей потребительской электроники. Linux становится доминирующей системой в области ПК, бытовых товаров и даже в автомобильной индустрии. Поэтому мы ожидаем от предстоящего события значительной отдачи.

LXF: Так-то оно так, но Asus недавно отступил с рынка Linux-нетбуков?

ДжЗ: Я думаю, положение со временем выправится. Это произошло из-за интервенции Microsoft с существенно (почти до нуля) уцененной XP. Не думаю, что они смогут проделать такой фокус с Windows 7. Надо же им как-то делать деньги!

LXF: Вероятно, пределы маневра на этих устройствах не так уж велики?

ДжЗ: Поле для маневра *обширное*, особенно для Microsoft. Предположим, Acer или Asus комплектуют ПК. Составляется смета расходов на изготовление. При розничной цене нетбука 250 долларов стоимость материалов составляет 150 долларов или что-то вроде, так что доля Windows составляет примерно треть.

LXF: Если Microsoft снова не сбросит цену до 5 долларов.

ДжЗ: Они вынуждены и будут пытаться получить 45 долларов с ПК. Об этом сообщалось в прессе (по-моему, в DigTimes). Ценообразование для OEM покрыто мраком, этот секрет свято охраняется. Сама Microsoft не любит об этом распространяться, да и OEM-партнеры помалкивают. Но 45 долларов для Windows – что-то вроде психологической отметки.

LXF: У вас есть новостной сайт Linux.com, Distribution Central с новостями и обновлениями для дистрибутивов, есть форумы, образовательные инициативы. Не кажется ли вам, что вы отжимаете с дороги сообщество, делая все в рамках Linux Foundation?

ДжЗ: Так ведь сообщество наш сайт и составляет. Весь контент (во всяком случае, основная его часть) Linux.com генерируется его пользователями. Их, кстати, более 7000...

LXF: Хорошо, но мы говорим о том, что прежде было много самостоятельных, независимых сайтов и форумов — посвященных отдельным дистрибутивам (форумы Ubuntu или SUSE) или более общим вопросам (LinuxQuestions.org). По части конференций есть весьма полезный OSCON или, например, Linux.conf.au...

ДжЗ: То есть мы вытесняем Linux Questions с его законного места?

LXF: Пожалуй, да! Вот цитата с вашего сайта, записанная нами: «Зачем изучать Linux где бы то ни было, если есть Linux Foundation»? Это слегка настораживает!

ДжЗ: Почему?

LXF: Потому что вы советуете людям не искать альтернативы. Проблема в том, что в вашей организации работают очень известные, авторитетные люди. Когда работодатели Линуса Торвальдса говорят: «никуда больше не ходи», как же тут не насторожиться?

ДжЗ: Пересмотрите свое мнение немедля! Это абсурд: в данной сфере множество обучающих организаций. Я думаю, мы обеспечиваем качество обучения не ниже, чем у остальных. Мы считаем, что рынок образовательных услуг для Linux необходимо расширять. Обучение не только приносит пользу людям, стремящимся глубже понять систему, но и служит средством финансовой поддержки организации.

Предлагаемые нами виды обучения не являются прямыми конкурентами других известных мне рыночных предложений: наши темы – повышение эффективности ядра или разработка драйверов для устройств. Да я первый скажу: «Если вы нашли курсы лучше наших, что вам мешает пойти туда учиться»? LXF

Люкс под Linux'ом

Если вам случится путешествовать из Петербурга в Москву на комфортабельном поезде, знайте: не в последнюю очередь вы обязаны этому Linux. **Олег Дмитриев** расскажет подробности.



Наш эксперт

Олег Дмитриев
Кто расскажет о бортовой Linux-системе лучше, чем человек, который ее обслуживает? Олег – системный администратор состава «Александр Невский».

Сотрудничество компании «ЖелДорСервис», в отделе информационных систем которой я работаю, с Российскими железными дорогами (РЖД) ведет свою историю с начала 2000-х годов. Тогда на поездах «ЭР200» и «Аврора» были установлены первые медиасистемы и системы видеонаблюдения, реализованные на оборудовании финского концерна Mitron OY (тогда он назывался Instavision Solution). Примечательно, что этим же концерном (и даже на том же заводе в городе Тампере) делалась НАТОвская авионика. Оборудование Mitron использует собственные протоколы передачи данных, что впоследствии подтолкнуло нас к переходу на Linux.

От Mitron к «Инбере»

Первые медиа-системы были простейшими. В каждый вагон монтировался некий источник сигнала (начинали еще с VHS-магнитофонов), который транслировался пассажирам. Это были аналоговые, нецентрализованные решения; все смотрели и слушали одно и то же, без возможности выбора. На вагон транслировались один видео- и два аудиопотока. Такая схема была принята как стандартная, и по ней были оснащены многие другие поезда.

Переход на Linux произошел после проекта «Red Arrow» (Красная Стрела, модернизация 2004–2005 гг). На составе была развернута полноценная IP-сеть для системы видеонаблюдения и установлена новая система с централизованной раздачей видеосигнала. Это был первый поездной видеосервер, транслирующий сигнал на весь состав. Мы использовали оборудование того же финского концерна Mitron; соответственно, видеосистема работала по своим протоколам (аналог RS-485).

После того, как стало ясно, что IP-сети прекрасно работают в «боевых условиях», было решено отказаться от дорогостоящего финского оборудования и перейти на них и Linux-системы. Linux был выбран не столько из-за цены, сколько из-за надежности и большей гибкости в настройке.

Нашим следующим Linux-проектом, после «Красной Стрелы», стал «Александр Невский». Это уникальный состав – действительно необычный поезд, на котором были реализованы все наши последние наработки. Появились доступ в Интернет, спутниковое телевидение, внутренние web-ресурсы... А главное – в отличие от ранних систем, появился выбор: пассажир мог сам решать, что смотреть, слушать, делать... В общей сложности, на состав установили 13 серверов, 80 тонких клиентов, две спутниковые тарелки и кучу сетевого оборудования – провели 100-Мбит-сеть на 15 вагонов. Все это дело три раза в неделю движется между Москвой и Санкт-Петербургом со скоростью 200 км/ч и неплохо работает. Так появилась первая бортовая Linux-система – «Инбера», используемая с сентября 2005 года. И сегодня я хочу рассказать о ней в подробностях.



› В стойке: внизу — сервер, далее — два ресивера, вверху слева — модуль управления спутниковыми тарелками, справа — маленький монитор (удобнее подключаться к системе с ноутбука).

«Александр Невский»

«Александр Невский» задумывался как туристический поезд, гостиница на колесах. Это очень большой состав: 18 вагонов, с собственной дизель-электростанцией, столовой, ресторанами и даже прачечной! Только для персонала отведено два вагона.

«Александра Невского» собирали на трех заводах: в Москве, Санкт-Петербурге и Воронеже. В период обкатки и как туристический состав он ходил в Астрахань, Монголию и Китай. Сегодня это поезд повышенной комфортности сообщением Москва–Санкт-Петербург. Изредка ходит в Хельсинки.



› «Александр Невский»: запуск состава в 2005 году.

По вагонам

В целом, «Инбера» – это не так уж и сложно. В пассажирских купе сейчас установлены в общей сложности 58 (от 1 до 4 в вагоне, по количеству купе) тонких клиентов CI852C (1 ГГц CPU, 512 МБ ОЗУ). В каждом купе, помимо терминала, есть розетка RJ-45, то есть можно подключить свой ноутбук и работать в Интернете. Купейный тонкий клиент на самом деле не такой уж и тонкий... Linux (мы используем Gentoo) стартует локально с Flash-накопителя и, загрузившись, «подтягивает» все, что нужно, с сервера. Поль-

зователь работает с системой через web-интерфейс: изначально на базе *Firefox 1.5*, сейчас – на основе *Firefox 2.0.0.20*. Через него пользователь получает доступ к ресурсам: музыке, кино, спутниковому телевидению... Для просмотра видео применяется подключаемый модуль VLC для Mozilla, хотя с ним не обошлось без проблем. Если нужно просто транслировать/принимать видеопоток, VLC – один из лучших. Но когда требуется организовать web-интерфейс управления потоком... Кнопки Play и Stop работают нормально, а вот Pause держит паузу недолго – настолько, что пассажир не успевает вернуться из туалета... Может, оно и смешно, а для нас это (наряду с перемоткой) было причиной одного из обновлений.

В каждом вагоне есть свой локальный сервер (P4 3,2 ГГц, 1 ГБ ОЗУ). Такая избыточность нужна на случай падения сети и при расцеплении вагонов – часть сервисов останется доступной. На серверах мы также используем Gentoo – его выбрал первый разработчик системы, Александр Тягов. Кстати, нынешняя команда такова: Николай Лысенко, Вадим Краснобаев, Игорь Матросов, Сергей Горький и я, Олег Дмитриев.

С вагонного сервера клиенты получают настройки: DHCP, DNS, NTP и так далее, но основная его задача – это web. Мы используем *Apache 1.3* в связке с PHP 5.2 и VLC 1.0.1 для видео. Создать аудиорешение на базе VLC по ряду причин не вышло, и мы выбрали *Icecast Streaming Server 2.3.2*. Такие серверы используют и для радиовещания в Интернете. Кроме того, на вагонных серверах хранятся сами медиа-файлы: MP3, Ogg, VOB. Мы называем их «консервами» – так уж повелось.

Головной сервер

Головной сервер находится в штабном вагоне. В радиорубке установлена небольшая стойка 19'', в которую вмонтированы сам сервер (P4 3,2 ГГц, 1 ГБ ОЗУ) и два ресивера (OpenBox CI7200PVR). Что характерно, ресиверы OpenBox тоже используют операционную систему Linux!

> «Что характерно, ресиверы OpenBox используют ОС Linux.»

Штабной сервер управляется Debian Etch; его основная задача – шлюз в Интернет. Единственный способ обеспечить доступ в Интернет на всем протяжении пути следования поезда – это спутник. На крыше вагона установлены две спутниковые тарелки TracVision C3: одна для Интернета, другая – для спутникового телевидения. Тарелки специализированные, ориентированные на использование с воды. У них есть своя система корректировки, и если поезд меняет направление, антенна тоже поворачивается, как бы постоянно держа спутник на прицеле. Спутниковый Интернет односторонний (асинхронный), то есть тарелка только принимает сигнал. Для запросов используется наземный шлюз, доступный через GPRS-канал. Мы используем промышленные GPRS-модемы Siemens MC75.

В последнее время наша команда столкнулись с проблемой нехватки уровня сигнала со спутника. В дороге она пока не решена, а в черте города выручает Yota и проект Madwimax, обеспечивающий работу Yota-модемов в Linux. Сейчас мы тестируем промышленное решение на базе WiMax – устройство MAX-306HW2. Оно состоит из двух компонентов – мощной внешней антенны и Wi-Fi маршрутизатора.

Коротко коснусь спутникового телевидения. Чтобы сохранить полосу пропускания, для трансляции спутниковых каналов в сеть используется технология группового вещания (multicast, см. врезку). Делается это аппаратно, с ресиверов OpenBox. На данный момент, мы транслируем два потока: обычно это новости и спортивный канал.

За централизованную раздачу сигнала пришлось заплатить страшную цену... Появились «тонкие места» – межвагонные соединения. Они рвутся, до них не так просто добраться, они сбоят после мойки, а что самое неприятное – их много. Справа и слева по бортам «Александра Невского» проложен кабель, и если рвется рабочая сторона, система автоматически переключается на резервную. Реализовано это на оборудовании фирмы Harting: в 2005 году, на момент построения сети, они как раз выпустили новый коммутатор с функцией переключения портов при обрыве линии. Проблема была в том, что он не поддерживал multicast. Мы собирались уже делать маршрутизацию на сервере, но разработчики (специально для нас!) выпустили новую версию прошивки. Разъемы и провода у Harting свои – на них выполнена магистраль, а разводка по вагонам осуществляется обычной витой парой.



› **Приделываем Yota-антенну на крыше поезда. Справа — спутниковые тарелки (ближняя — ТВ, дальняя — Интернет). Здание — ремонтное депо, Санкт-Петербург.**

На будущее

Все течет, все меняется – «Инбера» не исключение. В ближайшее время планируем перейти на сетевую загрузку клиентов, чтоб они были немного потоньше. На самом деле, это из-за проблем с электричеством: «флэшки» плохо переносят перепады напряжения. Возможно, будет реализована IP-телефония на базе протокола SIP. Но все это будет завтра, а пока «Инбера» на «Александре Невском» остается просто самой крупной, первой и единственной в своем классе бортовой информационная системой на РЖД. LXF

Что за штука... Multicast?

В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов. Так, адрес, входящий в диапазон 224.0.0.0–239.255.255.255 (класс D), считается групповым multicast-адресом, и обрабатывается маршрутизаторами особым образом. Если в пакете в качестве адреса назначения указан адрес класса D, то его должны получить все узлы, которым присвоен данный адрес. Эти узлы образуют multicast-группу. Один и тот же узел может входить в несколько multicast-групп. При этом узел (пользователь) сам определяет, в какие группы он входит, и оповещает об этом маршрутизатор.

Основное преимущество передачи данных по технологии Multicast в том, что при подключении к группе новых узлов не увеличивается нагрузка на канал. Например, наш видеосервер транслирует в сеть поток со скоростью 2 Мбит/сек. В обычной (Unicast) среде число потоков равнялось бы числу узлов, что при 50 клиентах полностью исчерпало бы ресурсы сети. А их 80, плюс Интернет, плюс видеонаблюдение... В Multicast-среде между сервером и активным маршрутизатором нужна фиксированная полоса в 2 Мбит/сек, что и позволяет добиться хорошего качества, не загружая при этом сеть.



Обслуживаем страницы с Lighttpd

Lighttpd – быстрая и легкая альтернатива Apache, нацеленная на сайты с большим трафиком. **Джульетта Кемп** покажет, что настроить его – раз плюнуть.

Apache сейчас стал стандартным web-сервером для большинства пользователей Linux, и это превосходный продукт. Но для некоторых весьма и весьма быстрых сайтов с высокой загрузкой он может отставать от идеала. Специально для них разработан *Lighttpd* – сервер, занимающий минимум памяти и работающий на максимуме скорости. Он особенно хорош для обслуживания статического контента, отлично масштабируется на несколько серверов и ладит с *FastCGI* (вы можете настроить *Lighttpd* в качестве оболочки для использования массива серверов *FastCGI*). Кроме того, как мы покажем на данном уроке, его легко настроить. Даже если ваш сайт не гонится за скоростью *Lighttpd*, поработать с ним приятно – он даже настраивается быстрее *Apache*!

> «Похоже, Lighttpd настраивается даже быстрее Apache.»

Как обычно, если вы желаете иметь самую свежую версию (1.4.24), загрузите исходные тексты с сайта проекта (www.lighttpd.net). Однако почти всегда лучше будет использовать пакет, имеющийся в дистрибутиве (поскольку это вариант с оптимальной поддержкой для длительного применения). В Debian/Ubuntu пакет называется *Lighttpd* (в Debian Lenny и Ubuntu 9.04 используется версия 1.4.19, что не очень и далеко от основного релиза проекта); доступен он и в виде RPM.

Меняем порт

Желаете протестировать *Lighttpd* совместно с существующим web-сервером *Apache* (или другим)? Измените строку **server.port** в **/etc/lighttpd/lighttpd.conf** на

```
server.port = 81
```

(начальную страницу придется сохранить в **index.lighttpd.html**, потому что в каталоге **/var/www** уже есть **index.html** от *Apache*). Кроме того, заметьте, что сервер работает на порту 81. Перезапустите *Lighttpd* командой **/etc/init.d/lighttpd restart** и проверьте: http://localhost:81.

Если у вас нет другого web-сервера, работающего по порту 80 (см. врезку внизу, если есть), то стандартные настройки, созданные при установке пакета, позволят вам запустить сервер немедленно командой **/etc/init.d/lighttpd start**. Если на время настройки вы хотите запустить его в режиме отладки (чтобы он находился в терминале и выводил служебные сообщения прямо на экран), скомандуйте **lighttpd -D -f /etc/lighttpd/lighttpd.conf**.

Перейдите на http://localhost, и перед вам предстанет базовая страница-заглушка (вид ее зависит от выбранного пакета). Базовая стандартная настройка, создаваемая при установке, выглядит как-то так:

```
# В вашем файле с настройками может быть больше строк,
причем в другом порядке: это нормально!
server.modules = (
  "mod_access",
  "mod_alias",
  "mod_accesslog",
  "mod_compress",
)
server.document-root = "/var/www/"
server.port = 80
server.errorlog = "/var/log/lighttpd/error.log"
accesslog.filename = "/var/log/lighttpd/access.log"
index-file.names = ( "index.php", "index.html",
  "index.htm", "default.htm",
  "index.lighttpd.html" )
url.access-deny = ( "~", ".inc" )
static-file.exclude-extensions = ( ".php", ".pl", ".fcgi" )
include_shell "/usr/share/lighttpd/create-mime.assign.pl"
```

Здесь все должно быть понятно! Представлено немножко модулей, включенных по умолчанию, настроены журналы (журнал доступа требует строки для модуля **mod_accesslog** в разделе **server.modules**), а также настроены имена индексных файлов (тех, которые отображаются при запросе http://пример.com/каталог/).

Параметр **url.access-deny** устанавливает, какие файлы вообще не должны обрабатываться: здесь это лишь файлы с суффиксом **~** (обычно так помечаются резервные копии, и вы вряд ли захотите, чтобы старые данные были доступны из Интернета), а также файлы **.inc**, являющиеся исходными текстами для генерации динамического содержимого. Заметим, что в данный момент наш web-сервер не работает с динамическим содержимым, поскольку нет подключенных и настроенных модулей; но даже после их настройки вам незачем выкатывать исходные тексты на всеобщее обозрение, поскольку это может вызвать проблемы с безопасностью.

Debian создает MIME-типы при помощи скрипта, который включен (и запускается) в строке **include_shell**; в других случаях вы можете указать их вручную (добавьте и другие строки, если хотите!):

```
# Замените строку include_shell этими строками
mimetype.assign = (
".html" => "text/html",
".txt" => "text/plain",
".jpg" => "image/jpeg",
".png" => "image/png"
)
```

Чтобы протестировать настройки перед стартом сервера, будет неплохо воспользоваться командой

```
lighttpd -t -f /etc/lighttpd/lighttpd.conf
```

Еще одно действие, которое не мешает выполнить заранее – создание пользовательских каталогов. Все просто:

```
lighty-enable-mod userdir
/etc/init.d/lighttpd force-reload
```

PHP и FastCGI

Сейчас ваш web-сервер работает только со статическим содержимым – и для этого *Lighttpd* особенно хорош. Есть даже крупные сайты, которые применяют его для своего статического содержимого, а для динамического берут альтернативу, потому что обслуживание статичных файлов куда проще. Однако *Lighttpd* отлично справляется и с динамическим содержимым, особенно по протоколу FastCGI, который предоставляет интерфейс между вашим web-сервером и внешними приложениями – он очень похож на CGI, но лучше масштабируется. FastCGI не зависит от платформы, поэтому любое FastCGI-приложение, уже установленное и настроенное в *Apache* или другом web-сервере, прекрасно заработает и в *Lighttpd*. FastCGI предназначен для борьбы с рядом ограничений производительности CGI, а *Lighttpd* улучшает его еще больше, делая внутреннюю нагрузку FastCGI сбалансированной, что увеличивает производительность по сравнению с *Apache* и **mod_php**.

Для настройки этого в вашем новом lighttpd-сервере необходим **mod_fastcgi**. Он должен находиться в **/etc/lighttpd/mods-available/**. Чтобы подключить его, воспользуйтесь

```
# /usr/sbin/lighty-enable-mod
# /etc/init.d/lighttpd force-reload
```

Скрипт *lighty-enable-mod* выдаст вам список доступных модулей и спросит, какие из них включить; после этого вы перезапустите сервер, чтобы применить изменения.

Мы рассматриваем запуск FastCGI с PHP (как и CGI, он не зависит от языка, поэтому вы можете использовать Perl или что-то еще). Тогда потребуется также добавить PHP5-CGI, если он у вас уже не установлен (в Debian Lenny и Ubuntu, это пакет *php5-cgi*).

Для настройки *php5-cgi* необходимо подправить файл **/etc/php5/cgi/php.ini**, добавив в его конец следующую строку:

```
cgi.fix_pathinfo = 1
```

Перезапустите сервер, и все готово: вы настроили *Lighttpd* для поддержки PHP через FastCGI! (К сожалению, пока что нет модуля, который писал бы за вас на PHP…)

Переписываем правила

Вам потребуется заново создать правила, которые у вас уже есть в файле **.htaccess**, поскольку *Lighttpd* его не использует. Вот пример переписывания правил под *Lighttpd*:

```
url.rewrite-once = ( "^/faq/([0-9]+)$" => "/faq.php?id=$1" )
```

Это преобразует http://www.example.com/faq/71 в http://www.example.com/faq.php?id=71.

Другой опцией является канонизация расширений файлов:

```
url.rewrite-once = ( "/(.*)\.htm" => "/$1.html" )
```

Причем те, кто использует **.htm** вместо **.html**, будут перенаправлены по корректному адресу. Чтобы включить перезапись, раскомментируйте строку **mod_rewrite** в разделе **server.modules** файла настроек и перезапустите сервер.

Настройка SSL

Другой полезной настройкой web-сервера является SSL, с целью доставки защищенных страниц по https://ссылка. В стандартном для Debian/Ubuntu *Lighttpd* SSL включен: желая удостовериться, что это так, выполните **lighttpd -v** и убедитесь, что выводится нечто подобное:

```
lighttpd-1.4.19 (ssl) - a light and fast webserver
```

(ssl) здесь главная часть. Затем воспользуйтесь **lighty-enable-mod**, чтобы включить SSL-модуль, как описано выше в разделе о PHP. Вам потребуется исправить файл **/etc/lighttpd/sitesavailable/10-ssl.conf**, введя корректные параметры сертификата вашего сервера:

```
$SERVER["socket"] == "10.0.0.9:443" {
  ssl.engine = "enable"
  ssl.pemfile = "/etc/lighttpd/www.server.org.pem"
  server.name = "www.server.org"
}
```

Необходимо изменить адрес в верхней строке, а также строки **ssl.pemfile** и **server.name**. Затем перезапустите сервер командой **/etc/init.d/lighttpd forcereload**, и вы сможете подключаться к вашему серверу по https://адрес.

Мы не будем здесь описывать настройку SSL-сертификатов сервера, поскольку это выходит за рамки данного урока – но очевидно, что для работы необходим действительный сертификат в месте, указанном в **ssl.pemfile**. Для создания сертификата изучите команду *openssl*.

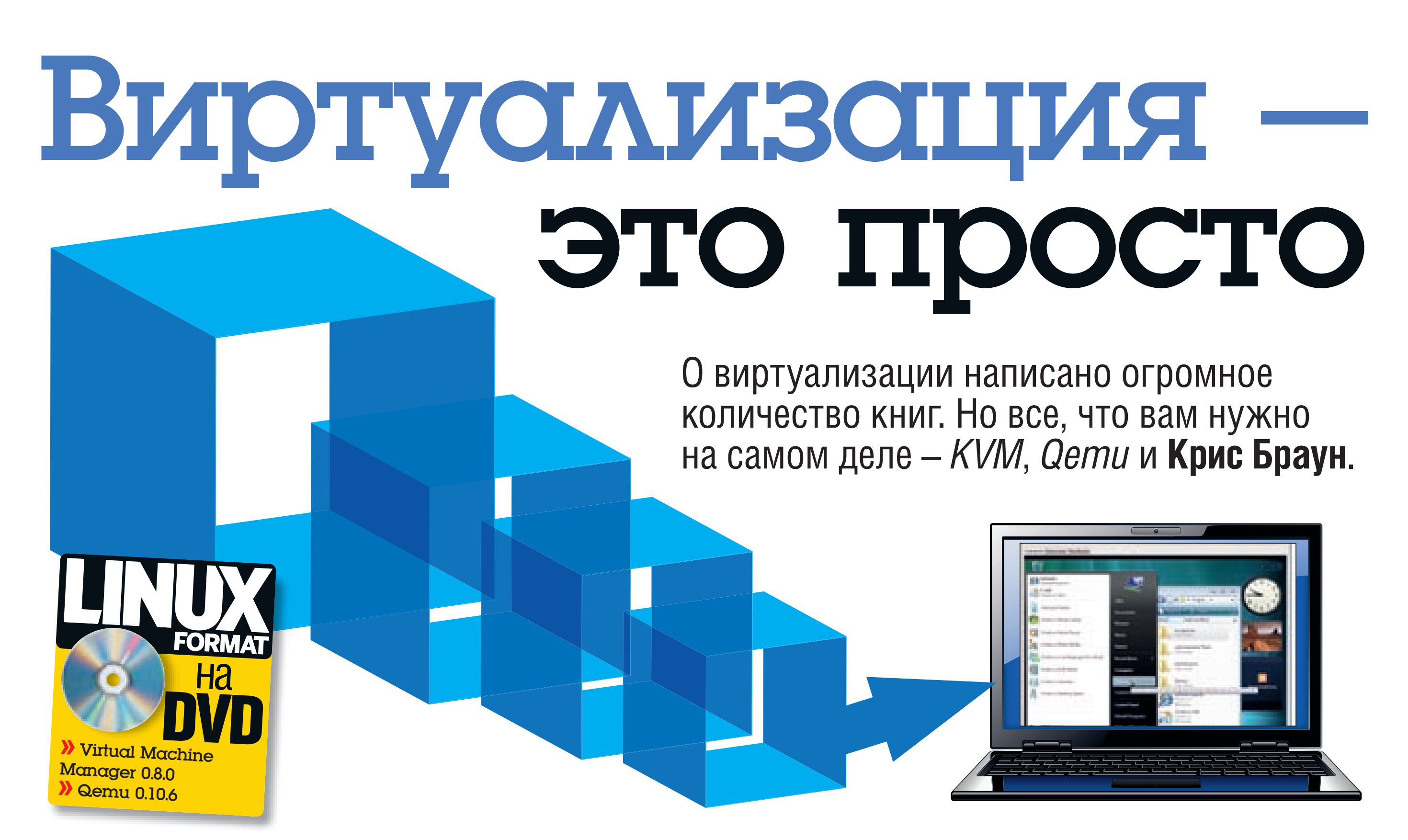
Настройка виртуального сервера

Lighttpd прекрасно поддерживает виртуальные серверы на базе URL, используемого для доступа к странице. Вы задаете выражение-условие в файле настроек, а *Lighttpd* определяет по нему, какое содержимое следует обрабатывать (и как его представлять). Вот пример настройки для создания виртуального хоста http://www.myothersite.net (чтобы все работало, необходимо, естественно, также иметь зарегистрированный и настроенный домен в DNS):

```
$HTTP["host"] == "www.myothersite.net" {
  server.document-root = "/var/www/myothersite/"
  $HTTP["url"] =~ "^/usefulscripts/" {
    dir-listing.activate = "enable"
  }
}
```

Если обрабатываемый URL содержит www.myothersite.net, *Lighttpd* воспользуется настройками внутри этого условия вместо стандартных для сервера. В данном случае, это означает смену корневого каталога сайта (то есть при запросе http://www.myothersite.net/test.html будет обработан файл **/var/www/myothersite/test.html**) и такую настройку каталога **usefulscripts**, чтобы можно было получать список содержимого директории (то есть чтобы запрос http://www.myothersite.net/usefulscripts/ выдавал перечень файлов в каталоге – изначально, по соображениям безопасности, это отключено). В условной конфигурации можно размещать любые опции и иметь сколько угодно виртуальных хостов. LXF

Виртуализация — это просто



О виртуализации написано огромное количество книг. Но все, что вам нужно на самом деле – *KVM*, *Qemu* и **Крис Браун**.

Согласно мудрой Wikipedia, «виртуализация — широкое понятие, обозначающее абстракцию ресурсов компьютера». Под это определение подпадает множество программных продуктов: *VirtualBox* от Sun, *Parallels*, Bochs, *Xen*, *KVM*, *Qemu*, разновидности *VMware* и многие другие. Есть и масса непонятных словечек, призванных сбить с толку неофита — эмуляция, полная виртуализация, паравиртуализация, виртуальное устройство, гипервизор… список можно продолжить. И не все придерживаются единого мнения о том, что эти термины означают.

Мы сознательно уйдем от копания в словах и взглянем с практической точки зрения на технологии виртуализации в Ubuntu, в частности, *KVM* и *Qemu*, и связанные с ними пользовательские утилиты для создания и управления виртуальными машинами (ВМ). Хотя мы будем говорить об Ubuntu, те же технологии применимы ко всем дистрибутивам Linux.

На рисунке слева я попытался показать некоторые из компонентов виртуализации в случае Linux. В нижней части рисунка представлены используемые технологии. Слева – *Xen*, программный слой, который концептуально лежит ниже ОС. Он называется гипервизором и запускается первым (после загрузчика) во время старта компьютера. *Xen* создает единственную виртуальную машину (Domain 0 или сокращенно Dom 0), на которой работает Linux. Именно этот домен предоставляет *Xen* свои виртуальные ресурсы. Дополнительные домены пользователя (Dom U) для дополнительных экземпляров Linux можно создать позже. Хотя мы и не будем больше о нем говорить, *Xen* – стабильное, мощнейшее решение для виртуализации. Canonical использует его, например, для ВМ Ubuntu, на которых студенты делают лабораторные работы по новому серверному курсу.

Пространство пользователя
virsh — libvirt
virt-viewer — libvirt
virt-manager — libvirt
libvirtd — libvirt
Гипервизор → Домен 0 (Гость), Домен U (Гость) — Xen
Модуль ядра KVM → Qemu (Гость, Гость) — KVM
Ядро → Qemu (Гость), Qemu (Гость) — Qemu

› **Некоторые из основных элементов виртуализации, и как они (примерно) соотносятся друг с другом, и с *libvirt* (в центре).**

KVM (не переключатель клавиатуры)

Компонент в средней части рисунка – *KVM* (Kernel Virtual Machine). *KVM* обеспечивает полную (аппаратно-зависимую) виртуализацию и может запускать неизмененные образы Linux или Windows, но для его работы необходимы соответствующие расширения процессора (Intel VT или AMD-V). Она состоит из загружаемого модуля ядра (**kvm.ko**), предоставляющего инфраструктуру виртуализации ядра и процессорного модуля (**kvm-intel.ko** или **kvm-amd.ko**, в зависимости от вашего процессора). Эти компоненты включены в основную версию ядра, начиная с 2.6.20. *KVM* использует *Qemu* для эмуляции устройств – таких как сетевая карта, жесткий диск, графический адаптер и т.д. Об этом сочетании мы здесь и поговорим.

В правой части рисунка мы видим *Qemu*, работающую в пространстве пользователя как чисто программный пакет эмуляции. Его можно применять и без всего (и при этом не нужны ни специальный модуль ядра, ни расширения виртуализации процессора, ни слой гипервизора), и на нем можно запускать неизмененные образы операционной системы. В самостоятельном

режиме *Qemu* – не самый быстрый вариант, однако его можно ускорить при наличии *KVM*, или с помощью модуля ядра *kqemu*.

В верхней части рисунка показаны некоторые инструменты пространства пользователя. *Virsh* – утилита командной строки для управления ВМ. Ее можно использовать для запуска, остановки, приостановки, возобновления, сохранения состояния и восстановления состояния ВМ или просмотра списка работающих ВМ. Она может подготовить ВМ к автоматическому запуску во время загрузки основного компьютера или даже к миграции на другую хост-систему.

Virt-viewer – инструмент для отображения графической консоли ВМ. Доступ к консоли осуществляется через протокол VNC.

Virt-manager – настольная утилита (написана на Python) для обслуживания ВМ. Она позволяет управлять жизненным циклом существующих машин (загрузка/выключение, пауза/возобновление, сохранение/восстановление состояния), созданием новых ВМ и виртуальными сетями, обращаться к графической консоли ВМ и просматривать статистику производительности.

Все эти утилиты опираются на *libvirt*, библиотеку C для взаимодействия с ВМ. *Libvirt* – настоящий герой нашей истории. Библиотека предоставляет стабильный, цельный API для управления машинами во множестве технологий виртуализации. На данный момент она, среди прочих, поддерживает *Xen*, *Qemu*, *KVM*, User Mode Linux и *VirtualBox*. Для определения виртуального оборудования *libvirt* применяет конфигурационные файлы в формате XML в каталогах **/etc/libvirt** и **/var/lib/libvirt**. Библиотека также используется демоном *libvirtd* – он является посредником во взаимодействии с системой виртуализации.

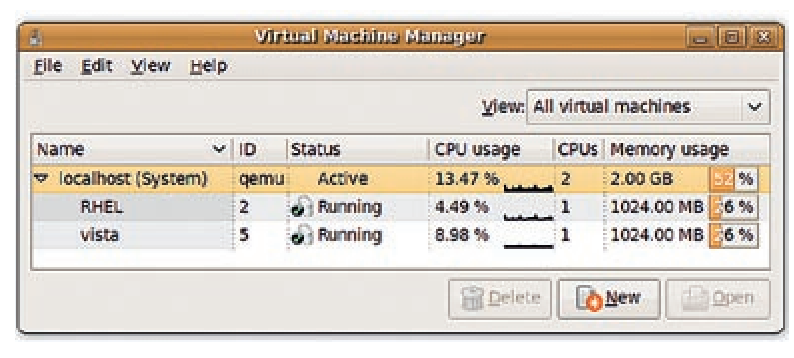
Ближе к делу

На нашем уроке мы создадим три ВМ на основе *KVM*. На первую установим Vista со стандартного носителя. На второй развернем Red Hat Linux с ISO-образа, а на третьей – Ubuntu напрямую из репозиториев.

Но сначала вот что: для работы *KVM* необходима поддержка виртуализации вашим процессором. Это означает, что требуются расширения процессора Intel-VT или AMD-V. Чтобы убедиться, поддерживает ли процессор одно из них, выполните команду

```
egrep '(vmxlsvm)' /proc/cpuinfo
```

Если команда ничего не вывела, процессор не поддерживает аппаратную виртуализацию. В противном случае – поддерживает, но нужно еще убедиться, что она активирована в BIOS.



› В главном окне *virt-manager* можно увидеть список ВМ, остановить или запустить их, открыть VNC-клиент для просмотра их консолей и узнать, как они используют процессор и память.

Избавившись от этого препятствия, применим *virt-manager* для создания виртуальной машины *KVM/Qemu* с Windows Vista. Для начала кликните на строке 'localhost' в главном окне *virt-manager* и выберите пункт меню New [Новый]. Последует серия окошек (некоторые из них показаны ниже), где нужно будет ввести разного рода информацию. Результатом этого процесса станут XML-файл, описывающий настройки виртуальной машины (в данном случае **/etc/libvirt/qemu/vista.xml**), и файл образа диска файловой системы ВМ (**/var/lib/libvirt/images/vista**). *Virt-manager* автоматически загрузит вновь созданную ВМ и запустит установку операционной системы с выбранного носителя.

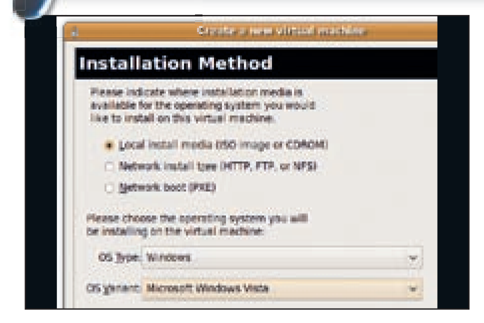
Образы ВМ можно создать и из командной строки, с помощью *virt-install*. Мы создали вторую ВМ и установили Red Hat из ISO-образа следующей командой:

```
$ sudo virt-install --connect qemu:///system --name RHEL5
--ram 1000 --disk path=/var/lib/libvirt/images/RHEL5.
img,size=8 --network network:default --accelerate --vnc
--cdrom /iso-images/RHEL5.2-x86_64.iso --os-type=linux
```

На создание ВМ ушло всего несколько секунд, и экран загрузки установщика Red Hat появился почти мгновенно. Установка и перезагрузка прошли без приключений.

Для третьей ВМ и в качестве альтернативы созданию пустой ВМ и установки ОС на нее мы воспользовались *VMBuilder*, скриптом на языке Python, умеющим создавать образ ВМ с нуля, на лету загрузив все его фрагменты из репозиториев Ubuntu. Это пере- »

Шаг за шагом: создаем виртуальную установку Vista



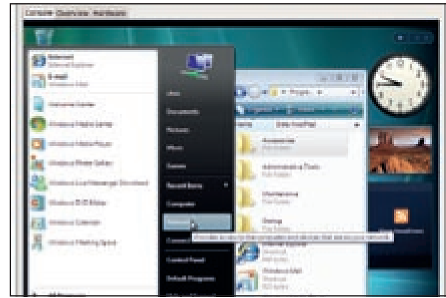
1 Выбираем источник

При запуске мастера создания ВМ вас попросят выбрать источник установки. В данном случае это физический компакт-диск Vista.



2 Создаем виртуальные разделы

Если у вас есть ISO-образ, с которого можно произвести установку, так и скажите. В противном случае воспользуйтесь CD-ROM'ом основного компьютера. Виртуальный диск может быть одним из физических разделов основного компьютера или файлом-образом в его файловой системе.



3 Выделяем ОЗУ

Укажите начальный и максимальный объем оперативной памяти, выделяемой ВМ, еще раз просмотрите всю информацию и наслаждайтесь результатом – Windows Vista, работающей на виртуальной машине *Qemu/KVM*.

деланный скрипт *ubuntu-vm-builder* из Ubuntu 8.04 LTS. Оба скрипта написал Сорен Хансен [Søren Hansen]. Вот пример:

```
$ sudo vmbuilder kvm ubuntu --suite jaunty --flavour virtual
--arch i386 -o --libvirt qemu:///system
```

Обязательны только два первых параметра. Здесь **kvm** задает гипервизор (т. е. технологию виртуализации, с которой работает образ). Поддерживаемые значения таковы: **esxi**, **xen**, **kvm**, **vmw6** и **vmserver**. Второй параметр, **ubuntu**, определяет дистрибутив, который вы хотите собрать. Сейчас поддерживается только Ubuntu. Время, необходимое для создания ВМ, в этом случае будет зависеть в основном от ширины канала между вами и репозиториями. Если вы планируете установить несколько похожих ВМ, обдумайте создание прокси-сервера репозиториев. Для этого достаточно выбрать один из компьютеров сети и установить пакет **apt-proxy**:

```
$ sudo apt-get install apt-proxy
```

Создастся APT-прокси с кэшированием, который слушает порт 9999. Чтобы направить *VMBuilder* на прокси, потребуется добавить опцию **--mirror** в его командную строку таким образом:

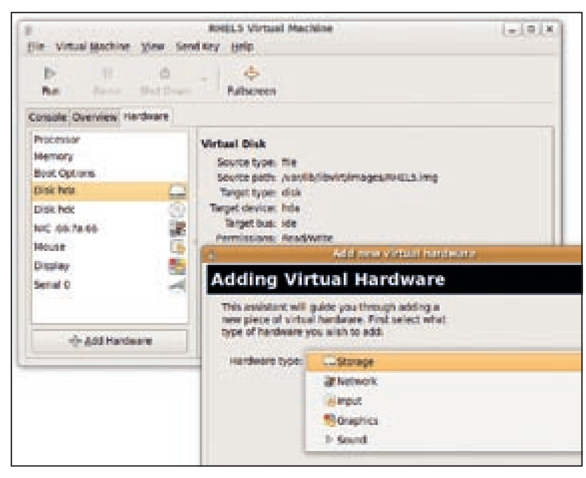
```
--mirror http://server-address:9999/ubuntu
```

Для первой созданной ВМ вы не заметите никаких изменений в быстродействии, так как прокси должен заполнить свой кэш из серверных репозиториев, но последующие должны работать гораздо быстрее (в простом тесте, время сборки для приведенной выше команды уменьшилось с 4 минут 36 секунд без прокси до 2 минут 26 секунд с прокси).

Если вам нужно получить несколько идентичных компьютеров, привлеките *virt-clone*. Эта утилита скопирует образы диска существующей ВМ и определит новый гостевой компьютер с той же конфигурацией виртуальных устройств, за исключением элементов, которые обязаны быть уникальными (например, MAC-адреса) и которые будут изменены во избежание конфликта. Например, мы создали клон виртуальной машины Ubuntu VM за какие-то секунды следующим образом:

```
$ sudo virt-clone --original ubuntu --name ubuntu3 --file /
home/chris/ubuntu-kvm/disk3.qcow2
Cloning from /home/chris/ 100%
|=========================| 364 MB 00:03
```

Для задания характеристик ВМ и установленного ПО у *VMBuilder* около пятидесяти параметров, поэтому вы почти наверняка захотите встроить команду в скрипт оболочки и пользоваться ею с удобствами. Некоторые из этих параметров приведены в таблице слева внизу.



› На вкладке Hardware [Оборудование] *virt-manager* можно просмотреть и изменить настройки ВМ и добавить новые устройства.

«Для задания характеристик ВМ у VMBuilder около 50 параметров.»

По умолчанию *VMBuilder* создает систему с минимальным содержимым. Дополнительные пакеты можно добавить, несколько раз указав опцию **--addpkg**. С помощью опции **--firstboot** задается скрипт, который будет запущен при загрузке образа, например, для установки сервера SSH, чтобы генерируемая им пара ключей была уникальной для каждой системы. Если надо установить пакеты, требующие при развертывании взаимодействия с пользователем (например, *MySQL* и *Postfix*), можно написать дополнительные скрипты, которые сделают это и будут запущены в первый раз, когда исходный пользователь (initial user) подключится к системе. На них указывает параметр **--firstlogin**. Такие возможности полезнее всего при создании образа виртуального устройства для передачи многим пользователям.

Чтобы не запускать *VMBuilder* с тысячей параметров, можно записать настройки во внешний конфигурационный файл и сослаться на него в командной строке следующим образом:

```
$ sudo vmbuilder kvm ubuntu -c myconfigfile.cfg
```

Какой бы вариант создания ВМ вы ни выбрали, в итоге у вас окажется два файла. Первый, в **/etc/libvirt/qemu**, представляет собой XML-файл, определяющий виртуальные устройства компьютера – его процессор, объем памяти, диски, к какой сети он подключен, его MAC-адрес и пр. Второй файл содержит образ диска компьютера (т. е. его файловую систему). *Virt-install* и *virt-manager* создают файлы с расширением **.img** и помещают их в каталог **/var/lib/libvirt/images**, *VMBuilder* создает файлы в формате QCOW2 и помещает их в ваш домашний каталог – в подкаталог, названный по гипервизору и дистрибутиву, для которого они были созданы. В нашем примере это **ubuntu-kvm**. 'cow' в имени файла – сокращение от Copy On Write (копирование при записи). В этом формате основной образ используется в режиме только для чтения, а сохраняются лишь различия. Это сэкономит много места на диске, если вы создаете несколько ВМ, большая часть файловых систем у которых общая.

Параметры VMBuilder

Параметр	Назначение
--mem=MEM	Выделяет виртуальной машине MEM МБ памяти
--rootsize=SIZE	Выделяет SIZE МБ корневому разделу
--swapsize=SIZE	Выделяет SIZE МБ разделу подкачки
--ip=ADDRESS	Устанавливает IP-адрес
--addpkg=PKG	Устанавливает PKG на виртуальную машину
--arch=ARCH	Задает архитектуру (AMD64 или i386)
--hostname=NAME	Устанавливает имя хоста в NAME
--mirror=URL	Загружает пакеты из репозитория по указанному URL
--lang=LANG	Устанавливает локаль в LANG
--user=USER	Имя пользователя initial user (по умолчанию Ubuntu)
--pass=PASS	Пароль initial user (по умолчанию Ubuntu)
--libvirt=URI	Регистрирует виртуальную машину в *libvirt* по заданному URI

Запуск виртуальной машины

Запустить виртуальную машину после установки можно разными способами. Низкоуровневый подход – вызвать *KVM* напрямую, просто передав ему образ файла виртуальной машины в качестве аргумента:

```
$ sudo kvm /var/lib/libvirt/images/RHEL5.img
```

Эта простая команда запустит ВМ с настройками *KVM* по умолчанию; например, у нее будет только 128 МБ ОЗУ (а не 1000 МБ, которые мы запросили при ее создании) и не будет никаких сетевых настроек. Теоретически это можно исправить, передав *KVM* соответствующие параметры, но проще запустить ВМ с помощью *virt-manager* или *virsh*. Они читают XML-описание ВМ и затем вызывают *KVM* с подходящими настройками.

Запустить ВМ с помощью *virt-manager* нетрудно. В главном окне кликните правой кнопкой на нужной виртуальной машине и выберите пункт меню Run (Запустить).

Virsh

Потом есть *virsh*. Удивительно, что без нее мы продвинулись так далеко, потому что *virsh* – это швейцарский нож в управлении ВМ. По сути, это обертка *libvirt* и консольный эквивалент *virt-manager*. При помощи *virsh* я могу запустить свою ВМ RHEL 5 таким образом:

```
$ sudo virsh -c qemu:///system start RHEL5
Connecting to uri: qemu:///system
Domain RHEL5 started
```

(Термин «домен» здесь просто означает «виртуальную машину». Он позаимствован из *Xen*, для которого *libvirt* изначально была написана.) Теперь можно просмотреть список запущенных ВМ:

```
$ sudo virsh -c qemu:///system list
Connecting to uri: qemu:///system
Id Name State
----------------------------------
1 RHEL5 running
```

При подобном запуске ВМ VNC-клиент автоматически не стартует, и если нужно увидеть ее консоль, запустите вручную утилиту просмотра:

```
$ sudo virt-viewer RHEL5
```

С помощью *virsh* можно просмотреть XML-определение домена (мы убрали часть строк ради экономии места):

```
$ sudo virsh dumpxml RHEL5
Connecting to uri: qemu:///system
<domain type='kvm' id='1'>
<name>RHEL5</name>
<uuid>2cadb958-6533-f728-e86a-421ec96ecfd3</uuid>
<memory>1024000</memory>
<os>
<type arch='x86_64' machine='pc'>hvm</type>
<boot dev='hd'/>
</os>
<devices>
<emulator>/usr/bin/kvm</emulator>
<disk type='file' device='disk'>
<source file='/var/lib/libvirt/images/RHEL5.img'/>
<target dev='hda' bus='ide'/>
</disk>
<disk type='file' device='cdrom'>
<target dev='hdc' bus='ide'/>
<readonly/>
</disk>
<interface type='network'>
<mac address='54:52:00:66:7a:66'/>
<source network='default'/>
<target dev='vnet0'/>
</interface>
<input type='mouse' bus='ps2'/>
<graphics type='vnc' port='5900' autoport='yes'
keymap='en-us'/>
</devices>
</domain>
```

Другие утилиты

Другие средства управления виртуальными машинами включают:

- **virt-viewer** Графическое приложение, подключаемое к консоли локальной или удаленной ВМ.
- **virt-clone** Утилита командной строки для клонирования образов существующих ВМ.
- **virt-image** Утилита командной строки для создания ВМ по XML-описанию образа.
- **virt-convert** Утилита командной строки для преобразования ВМ из одного формата в другой.
- **virt-top** Утилита, отображающая использование ресурсов ВМ в стиле *top*.

Возможно, вы захотите попробовать JeOS – урезанную версию Ubuntu Server с ядром, оптимизированным для запуска в виртуализированном окружении.

Файл **/etc/libvirt/qemu/RHEL5.xml** можно просмотреть и вручную. В нем подробно описана ВМ – ее название и UUID, объем выделенной памяти, необходимый для запуска эмулятор, расположение образа файловой системы, сеть, к которой она подключена, и многое другое. С помощью *virsh* также можно:

- Запускать, останавливать, приостанавливать и возобновлять ВМ.
- Переносить ВМ на другой компьютер.
- Создавать, удалять виртуальные сети, изменять их параметры и выводить их список.
- Управлять хранилищами образов ВМ.

Virsh принимает в общей сложности около 90 команд. Если вызвать ее без аргументов, *Virsh* запустится в режиме командного интерпретатора.



Библиотека *libvirt* — центральный компонент всего управления виртуальными машинами с помощью *Xen*, *KVM* или *Qemu*.

Поддержка сети

Виртуальные машины нужно подключать к виртуальным сетям. *Libvirt* позволяет создавать виртуальные мосты и объединять их в различные топологии. В установке по умолчанию *libvirt* создаст единственную сеть под названием 'default'. Ее XML-определение можно просмотреть с помощью *virsh*:

```
$ sudo virsh net-dumpxml default
Connecting to uri: qemu:///system
<network>
<name>default</name>
<uuid>a77f2c3f-e951-26c0-4c7d-eb23ace76051</uuid>
<forward mode='nat'/>
<bridge name='virbr0' stp='on' forwardDelay='0' />
<ip address='192.168.122.1' netmask='255.255.255.0'>
<dhcp>
<range start='192.168.122.2' end='192.168.122.254' />
</dhcp>
</ip>
</network>
```

Его можно просмотреть и напрямую: это файл **/var/lib/libvirt/network/default.xml**. В этом файле определена сеть **default**, занимающая диапазон IP-адресов 192.168.122.0/24. У хост-компьютера будет виртуальный интерфейс, подключенный к этой сети, с адресом 192.168.122.1. Сервис *dnsmasq* слушает его и предоставляет DHCP и DNS в этой сети. LXF

Чтобы узнать больше...

- Несколько хороших обзоров технологий виртуализации можно найти на сайтах http://www.ibm.com/developerworks/ru/library/l-virt/index.html, www.vmware.com/pdf/virtualization.pdf и http://virt.kernelnewbies.org.
- Подробное руководство по VMBuilder можно прочесть на https://help.ubuntu.com/community/JeOSVMBuilder.
- Libvirt описан на сайте http://libvirt.org.
- Как всегда, прочтите man-страницы для таких команд, как virt-install, virsh, kvmqemu и VMBuilder.

TaskJuggler: План через код



Есть ли у вас план проекта? **Алексей Джежора** подскажет, как создать его и отслеживать ход выполнения.



Наш эксперт

Алексей Джежора знает, что любую задачу можно решить тремя способами: сделать это самому, поручить другому или перенести срок ее выполнения на потом. На все должна быть «бумажка»!

Современное понимание ремесла управления проектами, будь то создание космического корабля, модернизация станка или строительство конуры для любимой собаки, немыслимо без применения специализированного программного обеспечения. И существующее свободное проектное ПО достигло такого уровня функциональности, что способно полностью вытеснить собой проприетарные продукты. Среди множества свободных программ (обзор и сравнение см. **LXF116**, а на самом деле их еще больше) по управлению проектами, своим нестандартным подходом выделяется *TaskJuggler*. На официальной странице (www.taskjuggler.org) можно найти исходный код для загрузки, много полезной информации, а также 200-страничное руководство пользователя в формате PDF на английском языке. Последняя стабильная версия 2.4.3 собрана в двоичные пакеты под многие дистрибутивы (Deb и RPM точно есть).

TaskJuggler ломает стереотип, сложившийся для программ по управлению проектами, и прежде всего потому, что план проекта со всеми зависимости буквально пишется во встроенном текстовом редакторе. При этом визуализирующие отчеты, получаемые посредством TaskJugglerUI, могут формироваться из нескольких файлов, создавать которые могут разные специалисты. Все это, на самом деле, очень удобно: экономится куча времени, нет необходимости щелкать мышью в бесконечном количестве диалогов, а отчеты хорошо оформлены и их, так сказать, не стыдно и подчиненным, и начальству, и заказчикам показать.

Первые шаги

Сегодня мы составим упрощенный план проекта создания автоматизированной информационно-измерительной системы технического учета электрической энергии (АИИС ТУЭ) приемо-сдаточного пункта. АИИС – это совокупность программно-аппаратных средств, которые измеряют какие-либо величины, результаты этих измерений обрабатывают, хранят и оправляют по назначению.

Интерфейс *TaskJuggler* интуитивно понятен и состоит из меню, панели инструментов, вкладки текстового редактора Editor и вкладки отчетов Reports. Для создания плана нового проекта нажмите Ctrl+N (или выберите пункт File > New, или просто щелкните по кнопке в форме листа бумаги на панели инструментов). В возникшем окне укажите папку, где сохранится ваш файл, и имя файла. Проследите, чтобы в полном пути папки (включая каталоги вышестоящих уровней) и самом названии файла были только латинские символы, иначе *TaskJuggler* откажется работать. Файл нашего проекта назовем **Avtomatizatsiya**, расширение у него будет **.tjp**. Вам предложат три исходных шаблона для проекта: очень простой, посложнее и очень сложный. Воспользуемся простым, именуемым Blank Project. В текстовом редакторе появится:

```
project myProjectId “My Project” “1.0” 2009-10-13 - 2010-12-11
task start “Project Start” {
  start 2009-10-13
}
```

Интервал дат начала и конца, разумеется, будет зависеть от даты непосредственного создания плана проекта в *TaskJuggler*.

Создаем свое

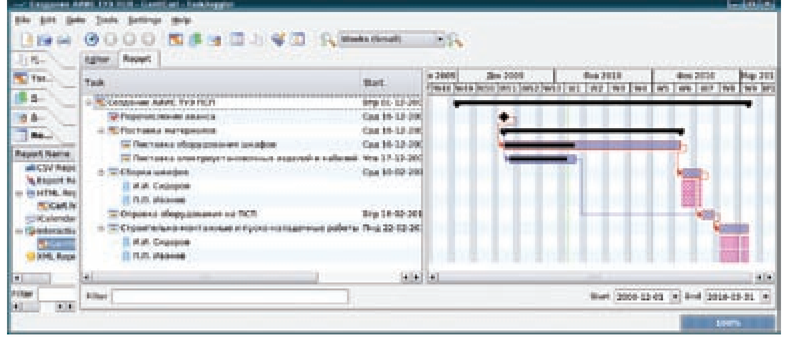
Приступим к планированию: назовем наш проект, укажем формат даты, создадим суммарную задачу проекта. Заменим код, приведенный выше, на

```
project myProject “Создание АИИС ТУЭ ПСП” “1.0” 2009-12-01+4m {
  timeformat “%Y-%m-%d”
}
task sum “Создание АИИС ТУЭ ПСП” {
  start 2009-12-01
}
```

Так как в процессе планирования удобно периодически смотреть, что получается, то сразу создадим соответствующий отчет: таблицу с колонками «индекс иерархии», «наименование», «начало», «окончание», «длительность» и ассоциированную с этой таблицей диаграмму Гантта:

```
taskreport “GanttCart” {
  headline “Создание АИИС ТУЭ ПСП”
  columns hierarchindex, name, start, end, duration, chart
  timeformat “%a %d-%m-%Y”
  loadunit days
}
```

Нажмем F9 (или, если еще есть ностальгия по работе мышью, выберем пункт меню Tools > Generate All Reports, или кнопку на панели инструментов в форме часиков): откроется вкладка отчетов с таблицей из одной строки и пустой диаграммой Гантта. Обратите внимание на формат даты: в *TaskJuggler* он имеет вид Год-Месяц-Число-Час:Минута:Секунда-ЧасовойПояс. Обязательными к заданию являются год, месяц и число, а все остальное указывается при необходимости. В отчетах такой порядок отображения дат можно менять, а также добавлять «подробности», например, ДеньНедели Число-Месяц-Год. А вообще, *TaskJuggler* поддерживает более сорока типов календарных и временных характеристик, включая даже номер века.



› Отчет «Диаграмма Гантта» — стандартный способ слежения за ходом выполнения проекта.

Планируем

Теперь приступим к составлению плана. Для начала опишем сотрудников, которые будут исполнителями работ по проекту. В нашем случае это инженер Петр Петрович Иванов, специалист по снабжению Сидор Сидорович Петров и электромонтажник 5-го разряда Иван Иванович Сидоров. Описание поместим перед суммарной задачей проекта:

```
resource people "Люди" {
  resource ppl_PPI "П.П. Иванов"
  resource ppl_SSP "С.С. Петров"
  resource ppl_IIS "И.И. Сидоров"
}
```

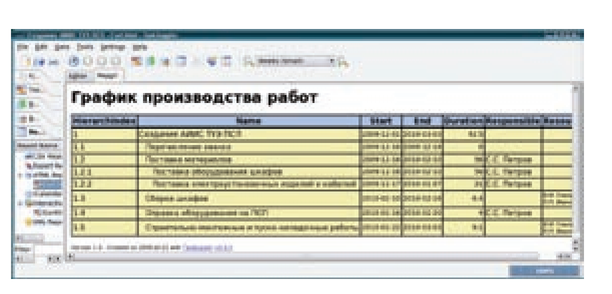
Затем «внутри» суммарной задачи (в промежутке между строкой со **start** и строкой с символом '**}**') пропишем последовательность работ, с зависимостями, назначениями и отслеживанием через процент выполнения. Начнем с получения аванса и поставки материалов:

```
task avans "Перечисление аванса" {
  start 2009-12-16
  milestone
}
task postavka "Поставка материалов" {
  responsible ppl_SSP
  task po_skaff "Поставка оборудования шкафов" {
    depends !!avans
    duration 8w
    complete 40
  }
  task po_electro "Поставка электроустановочных изделий и кабелей" {
    depends sum.avans {gaplength 2d}
    duration 3w
  }
}
```

Задача avans обозначена как веха – **milestone**: в предшествующей зависимости от нее находятся задачи по поставке материалов. Эта зависимость задается оператором **depends**. Ссылки на определяющую задачу могут обозначаться различными способами: символом '!', если эта задача находится в пределах одного уровня подзадач, символами '!!', если в пределах вышестоящего уровня подзадач, или может быть прописана глобальная ссылка, как в задаче po_electro. Через **gaplength** задается запаздывание начала задачи. Оператором **complete** определяется процент выполнения; если его не указывать, то *TaskJuggler* самостоятельно определит его как отношение между длительностью интервала времени от даты начала задачи к текущей дате и длительностью задачи в целом.

О назначениях ресурсов

За поставку материалов оператором **responsible** назначен ответственный, а оператором **duration** задана длительность выполнения задач. Если же представить себе, что С.С. Петрову придется самому ехать туда, где делают оборудование шкафов, и с этим оборудованием возвращаться обратно, а потом отправиться за кабелями и электроустановочными изделиями, то вместо длительности логично будет задать трудоемкость – **effort**, и оператором **allocate** назначить исполнителя (это тоже будет Сидор Сидорович). Длительность выполнения задач будет рассчитана, исходя из трудоемкости и загруженности назначенного(-ных) исполнителя(-лей). *TaskJuggler* распределит график выполнения так, чтобы не возникало конфликта назначенных ресурсов. Если назначать ресурсы некорректно, например, указать фиксированные пересекающиеся даты начала и конца выполнения задач (операторы **start** и **end**) с одними и теми же исполнителями, то в отчетах можно будет видеть визуализацию их перегрузки. Это также касается и материальных ресурсов. При этом *TaskJuggler* может работать с общекорпоративным списком ресурсов, что теоретически делает это приложение системой многопроектного планирования уровня среднего предприятия.



› HTML-отчет можно открыть в браузере; данные из него доступны для копирования в другие приложения — скажем, в *OOo*.

Завершим планирование нашего проекта, определив следующие задачи:

```
task sb_skaff "Сборка шкафов" {
  depends !postavka.po_skaff
  effort 2w
  allocate ppl_PPI, ppl_IIS
}
task otpr_mtr "Оправка оборудования на ПСП" {
  depends !postavka.po_electro, sum.sb_skaff
  duration 4d
  responsible ppl_SSP
}
task cmr_pnr "Строительно-монтажные и пуско-наладочные работы" {
  depends !otpr_mtr
  effort 3w
  allocate ppl_PPI, ppl_IIS
}
```

Все использованные конструкции уже должны быть вам знакомы.

Чтобы было красиво

Теперь, помимо уже созданного отчета с диаграммой Гантта, сформируем HTML-отчет – таблицу «График производства работ», с дополнительными колонками «ответственный» и «ресурсы»:

```
htmltaskreport "Cart.html" {
  headline "График производства работ"
  columns hierarchindex, name, start, end, duration, responsible, resources
  loadunit days
}
```

Нажмем F9 – и... вуаля! Остается только распечатать и нести на утверждение. LXF

Вести с полей

При помощи *TaskJuggler* планировался и в настоящее время воплощается проект создания автоматических информационно-измерительных систем коммерческого и технического учета электрической энергии с элементами управления энергохозяйством (АИИС КУЭ, АИИС ТУЭ) на приемо-сдаточном пункте нефтепровода «Сковородино – граница КНР». Работы ведет ООО НПК «Спецэлектромаш» (г. Красноярск). Дело в том, что у руководителя этого проекта и, по случаю, автора этой статьи, на рабочем компьютере «навернулась» файловая система одной из проприетарных ОС. Было решено пойти на смелый эксперимент: установить Linux. Выбор пал на Runtu 3.0. Практически сразу встал вопрос о ПО для управления проектами. Первоначально были попытки применять схожие по функциональности с одной известной проприетарной программой средства, но все они обладали существенными недостатками: *Planner* и *KPlato* подпадали под неразрешимые зависимости, *OpenProj* ел вычислительные ресурсы компьютера как пончики и безнадежно «тормозил». *TaskJuggler*, во-первых, не обладал подобными недостатками, а во-вторых, оказался очень удобным. Итоговый план проекта рассчитан на период более года и содержит более пятидесяти задач с назначениями ответственных за их исполнение.

RPM: Нынче не то, что давеча

Если вы до сих пор считаете RPM пакетной системой второго сорта, **Алексей Федорчук** попробует переубедить вас, заодно затронув новое веяние в этой области – *PackageKit*.



Наш эксперт

Алексей Федорчук
Перепробовал большую часть Unix-систем, выходивших за последние ...дцать лет, и пришел к выводу, что RPM и Fedora его вполне устраивают.

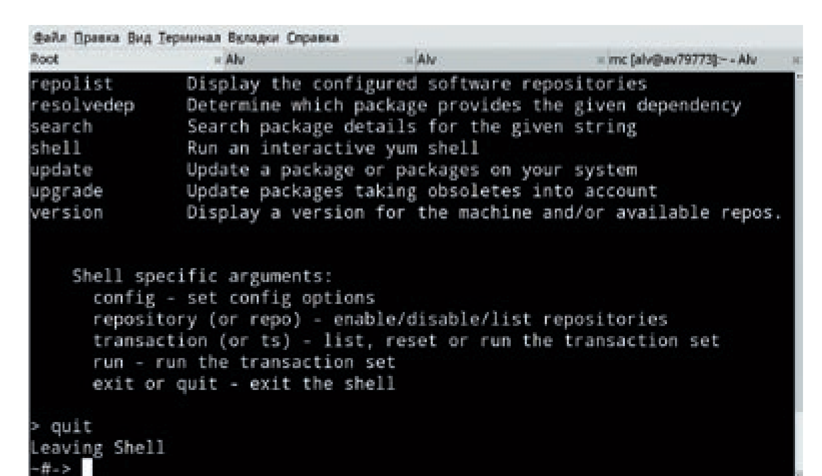
Многие, чье знакомство с Red Hat и его клонами пришлось на 90-е годы, надолго сохранили предубеждение и против формата их пакетов, и против утилиты управления оными. Конечно, написать rpm -ihv проще, нежели собрать нужный пакет из исходников. Однако в сравнении с портами FreeBSD с одной стороны и *APT* от Debian с другой, это выглядело бледно. Но все течет, все меняется – и ныне RPM-дистрибутивы располагают развитыми системами пакетного менеджмента, работающими как в текстовом, так и в графическом режиме. В настоящей статье мы остановимся на двух из них – *Yum* и *PackageKit* – на примере дистрибутива Fedora.

Часть 1: Незнаменитый Yum

Yum – система управления RPM-пакетами и их репозиториями, предлагающая автоматическую установку, обновление и удаление пакетов и пакетных групп с контролем зависимостей. По механизму действия и функциональности она сходна с *APT*. Однако если последний получил широкую известность – не в последнюю очередь благодаря популярности Ubuntu, а также тому, что усилиями сначала Connectiva, а затем ALT Linux широко распространился за пределами родного дистрибутива, то *Yum* остается сравнительно малоизвестным.

> «Yum способен сделать конфетку из не самого приятного продукта.»

По своим возможностям *Yum* для RPM ничуть не уступает утилитам *APT* для Deb, и используется достаточно широко: эта система принята в качестве основной в Fedora, RHEL и их прямых и косвенных потомках.



› **Yum и его интерактивная оболочка.**

Yum означает Yellow Dog Updater, Modified, то есть Обновитель Yellow Dog Модифицированный. Однако его связь с одноименным дистрибутивом – портом Red Hat на архитектуру Power PC – не совсем прямая. Просто пакетный менеджер Yellow Dog, *YUP*, послужил основой, на которой Сет Видал [Seth Vidal] писал *Yum* для дистрибутива Red Hat. Дословный перевод названия (англ. «ням-ням») можно трактовать и так, что *Yum* способен сделать конфетку даже из такого… не самого приятного продукта, как пакеты в формате RPM.

Yum быстро получил признание среди ряда клонов Red Hat – в частности, был принят в качестве штатного менеджера пакетов в ASPLinux. Однако в самом Red Hat он долго конкурировал с *apt-rpm*, и развитие *Yum*'а одно время только силами команды ASPLinux и осуществлялось. Однако в конце концов он утвердился в RHEL и его клонах (CentOS, Scientific Linux), в Fedora и в Yellow Dog.

Система *Yum* (yum.baseurl.org) включает собственно одноименную утилиту, набор дополнительных инструментов (*yum-utils*) и многочисленные дополнения, расширяющие функциональность основной программы.

Запускается *Yum* командой **yum**, требующей указания субкоманды (возможно, с опциями) и, в ряде случаев, аргументов в виде имени пакета или группы пакетов, что в общей форме выглядит так:

```
$ yum subcommand [arguments] --[options]
```

Без указания субкоманды *Yum* выведет краткую справку касаемо последних и их опций. Аналогичный результат дает

```
$ yum help
```

А указание имени субкоманды в качестве аргумента **help**, например,

```
$ yum help install
```

выведет краткие сведения о ее назначении.

Азбука сиинтаксиса

Субкоманды *Yum* определяют действие, которое нужно выполнить – установку или удаление пакета, вывод информации о нем, поиск и так далее. Обычно назначение субкоманды легко угадывается из ее названия и (или) краткой характеристики в выводе **yum help**.

Субкоманды *Yum* можно разделить на две группы. Первая связана с поиском пакетов и получением сведений о них, вторая – с манипуляциями пакетами и группами.

В составе первой группы наиболее употребимы:

» **search [строка]** Поиск пакета по имени или его фрагменту.

» **list** Вывод списка пакетов: всех (all или без указания фильтра), установленных (installed) или доступных (available).

» **info имя** Вывод полной информации о пакете.

Все субкоманды первой группы могут выполняться от лица обычного пользователя.

Во второй группе субкоманд наиболее важны:

» **install пакет1 пакет2 ...** Установка из репозиториев одного или нескольких пакетов, имена которых (в краткой форме) даны в качестве аргумента, вместе со всеми их зависимостями.

» **localinstall путь/к/пакету.rpm** Установка пакета из локального файла; зависимости извлекаются из репозиториев, если таковые доступны.

» **update [имя]** Обновление пакета, указанного в качестве аргумента; в отсутствие аргумента выполняется тотальное обновление системы, аналогично **apt-get update** и **apt-get upgrade**.

» **erase pkgname** Удаление пакета вместе со всем, что от него зависит; пакеты, от которых зависит удаляемый, остаются в неприкосновенности, даже если они никем не используются.

Субкоманды второй группы требуют наличия прав администратора.

Отдельно надо сказать о субкоманде **shell** – она запускает собственную интерактивную командную оболочку *Yum*, в сеансе которой можно оперировать субкомандами, аргументами и опциями, опуская главную команду **yum**.

Исполнение любой субкоманды начинается с синхронизации локальной базы пакетов с базами репозиториев. Затем происходит проверка зависимостей – и по ее результатам выводится итог: сколько пакетов, включая зависимости, должно быть установлено, обновлено или удалено; их имена; подлежащий скачиванию объем информации. Все завершается подтверждением на выполнение операции.

В состав пакета *yum-utils* входит серия утилит, запускаемых как самостоятельные команды, со своими опциями. Полный их список можно получить из

```
$ man yum-utils
```

Важнейшей из этих команд является **package-cleanup**, предназначенная для получения сведений о неполадках в локальной базе данных пакетов и их устранения. Она имеет несколько опций. Например,

```
$ package-cleanup --problems
```

выведет список нарушенных зависимостей, а с помощью команды

```
$ package-cleanup --leaves
```

можно вывести список пакетов, от которых не зависят никакие другие.

Дополнения, в отличие от утилит, как самостоятельные команды не запускаются, а встраиваются в команду **yum**, добавляя ей новые функции. Например, в RFRemix по умолчанию устанавливаются следующие расширения:

» **fastestmirror** Проверка скорости доступа к зеркалам репозитория и выбор самого быстрого из них; выполняется при каждом запуске *Yum*.

» **presto** При обновлении пакетов скачивает из репозиториев только изменения (deltarpms), минимизируя таким образом трафик.

» **refresh-packagekit** Обеспечивает обновление системы *PackageKit*, о которой мы поговорим ниже.

Тонкий тюнинг

Эффективное использование *Yum* требует некоторых мероприятий по настройке, включающих

» настройку собственно *Yum*;

» подбор и настройку дополнений;

» подключение дополнительных репозиториев.

За первый пункт отвечает файл **/etc/yum.conf** – он содержит общие для этой утилиты параметры в формате

```
название=значение
```

Значение может быть булевым (0 – запрещено, 1 – разрешено), численным – от 1 и до... разумного предела (0 равносильно отключению), или символьным – например, путем к каталогу или списком пакетов; в последнем случае значения разделяются пробелами. По умолчанию установлено следующее:

» **cachedir=/var/cache/yum** Каталог для кэширования метаданных репозиториев и пакетов, скачиваемых в ходе установки.

» **keepcache=0** Определяет, сохранять ли скачанные пакеты в локальном кэше или удалять их после успешной установки.

» **debuglevel=2** Уровень отладочных сообщений.

» **logfile=/var/log/yum.log** Каталог для файлов протоколирования действий *Yum*.

» **exactarch=1** Устанавливать пакеты, точно соответствующие архитектуре.

» **obsoletes=1** Определяет логику замены «устаревших» пакетов при тотальном обновлении.

» **gpgcheck=1** Проверять подписи пакетов при установке.

» **plugins=1** Использовать дополнения.

» **installonly_limit=3** Максимальное количество пакетов, запрещенных к обновлению (можно только устанавливать более новую версию параллельно).

Существует еще немало параметров настройки *Yum* помимо перечисленных. Так, очевидно, что опция **installonly_limit** имеет смысл только при наличии списка запрещенных к обновлению пакетов. Он задается параметром

```
installonlypkgs=пакет1 пакет2 ...
```

Есть возможность и задать список пакетов, для которых запрещено как обновление, так и инсталляция, что иногда требуется при использовании проприетарных пакетов:

```
exclude=пакет1 пакет2 ...
```

Полезным может оказаться **skip_broken** – он заставляет пропускать установку пакетов с нарушенными зависимостями. Параметр **recent** нужен для субкоманды **list** с одноименной опцией: он устанавливает срок, в течение которого добавленные в репозиторий пакеты считаются новыми.

Что очень раздражает в *Yum*, так это синхронизация метаданных о репозиториях, происходящая каждый раз при его запуске с любой субкомандой – даже от лица пользователя, когда реально кэш метаданных обновлен быть не может. Такая ситуация изменяется параметром **metadata_expire**, которому можно дать то значение, которое покажется разумным. Или вписать строку

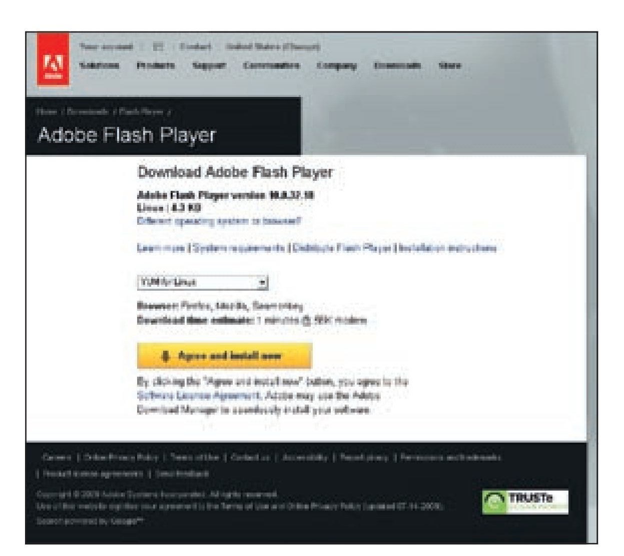
```
metadata_expire=never
```

и тогда обновление кэша метаданных будет производиться только по запросу.

Обратимся к дополнениям. Устанавливаются они точно так же, как и любые другие пакеты. Соответствующие каждому из расширений конфигурационные файлы находятся в **/etc/yum/pluginconf.d** и имеют говорящие имена. Большинство таких файлов предельно просто и содержит единственную строку, разрешающую подключение дополнения:

```
enabled=1
```

Но в настройках *Presto*, например, можно запретить локальное кэширование дельт, раскомментировав параметр »



› Adobe предоставляет *Flash Player* через репозиторий, но сначала его надо подключить.

keepdeltas = false

А можно определить, что считать дельтой. Например, параметр

minimum_percentage = 95

указывает, что если измененная часть пакета составляет 95 % или менее от цельного, то будет скачиваться она, если же больше – загрузится пакет целиком.

Новые репозитории

Чтобы настроить параметры доступа к репозиториям, их необходимо сначала подключить. Это не сложно: вся метаинформация о любом репозитории, пригодном для *Yum*, собрана в виде обычного RPM-пакета, который можно просто установить. Загвоздка в том, что этот пакет хранится внутри собственного, еще не подключенного, репозитория, и потому через *Yum* добавлен быть не может. Придется скачать его вручную, установить командой **rpm**, а затем уже обеспечить доступность репозитория.

Рассмотрим эту процедуру на примере подключения репозитория для пакетов проигрывателя *Adobe Flash*. Для этого заходим на официальный сайт Adobe (www.adobe.com), в пункте Download отыскиваем строку Get Flash Player, и из выпадающего списка Select version to download… выбираем YUM for Linux, который и скачиваем (в виде файла **adobe-release-i386-1.0-1.noarch.rpm**). Затем даем команду

```
# rpm -Uhv adobe-release-i386-1.0-1.noarch.rpm
```

По ее успешном выполнении, в каталоге с настройками репозиториев можно будет увидеть новый файл **adobe-linux-i386.repo**. Одновременно он станет доступным для обновляющих манипуляций командой

```
# yum update
```

Подключить новый репозиторий можно и совсем вручную. Проделаем эту операцию для репозитория (почти) ежедневных сборок браузера *Chromium* от Тома Коллуэея [Tom Callaway]: создадим в каталоге **/etc/yum.repos.d** файл **chromium.repo** и впишем в него такие строки:

```
[chromium]
name=Google Chrome
baseurl=http://spot.fedorapeople.org/chromium/F$releasever/
enabled=1
gpgcheck=0
```

Надеюсь, мне удалось показать, что *Yum* делает употребление RPM-пакетов абсолютно безвредным. В случае же напряженных отношений с командной строкой для управления RPM-пакетами можно обратиться к графической утилите *PackageKit*, к которой мы и переходим.

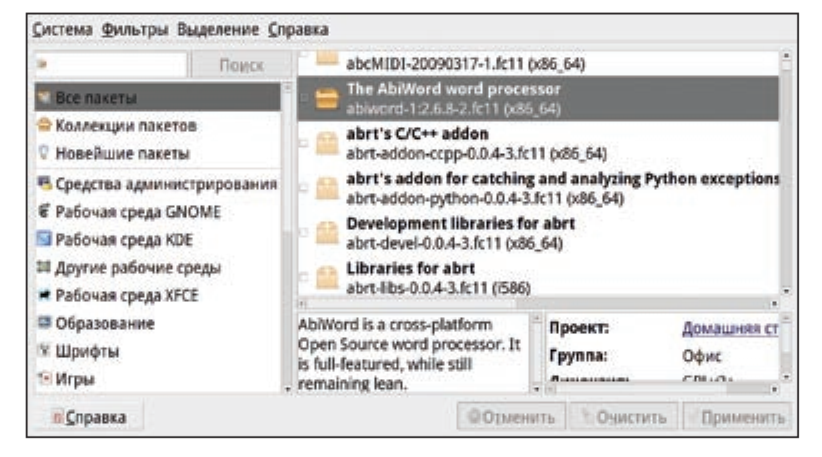
Часть 2: PackageKit — кит пакетного менеджмента

Если *Yum* всегда оставался в тени *APT*, то о надстройке *PackageKit* (www.packagekot.org) говорят еще меньше. Хотя она не является чем-то специфическим для RPM-дистрибутивов: ее можно приспособить к чему угодно и любым Linux-системам, вплоть до Arch и Gentoo.

Система *PackageKit* распадается на серию «драйверов» [back-end] для работы с конкретными менеджерами пакетов и интерфейсные надстройки. Драйверы *PackageKit* поддерживают такие инструменты, как *Yum*, *APT*, *Smart* и так далее, вплоть до *Pacman*. Интерфейсом к ним служат либо консольная утилита *pkcon*, одинаковая во всех дистрибутивах и в отношении синтаксиса команд не зависящая от нижележащего пакетного менеджера, либо графические оболочки *gnome-packagekit* и *KPackageKit* для Gnome и KDE соответственно.

При инсталляции в Fedora по умолчанию устанавливается драйвер для *Yum* и оболочка *gnome-packagekit* (при выборе в качестве рабочей среды KDE он заменяется на *KPackageKit*). В репозиториях доступны пакеты поддержки *APT* и *Smart*, а также консольный клиент *pkcon*.

Пакетные менеджеры, поддерживаемые системой *PackageKit*, имеют обычно собственный развитый инструментарий для управления пакетами из командной строки (*Yum* и Fedora, как мы видели, не исключение). Поэтому консольная утилита *pkcon* представляет интерес только своей теоретической универсальностью – она одинакова во всех дистрибутивах, поддерживающих *PackageKit*; так что задерживаться на ней не будем.



› PackageKit — общий вид. Пароль root запрашивается по ходу процесса.

Приятный интерфейс

Графическая ипостась *PackageKit* в виде субпакета *gpk-application* запускается из стартового меню, в зависимости от используемой среды, через пункты Приложения > Установка и удаление программ (Gnome) или Администрирование > Установка и удаление программ (*Xfce*). Причем сделать это можно от лица обычного пользователя – пароль администратора будет запрашиваться по ходу дела, при необходимости выполнения действий, требующих соответствующих полномочий. После запуска перед нами появляется окно, показанное на рисунке.

Переключаясь на соответствующие пункты в левой части, в правой вы будете видеть список всех пакетов – как установленных, так и доступных в репозиториях. Списки пакетов и коллек-

ций можно фильтровать по статусу (установлен или доступен), назначению (для разработчиков или конечных пользователей), режиму (графический или текстовый) и степени свободы (free или non-free). По умолчанию никакая фильтрация не производится.

Свободное поле с кнопкой Find [Поиск] рядом прямо так и провоцирует выполнить поиск некоего пакета. Он осуществляется по совпадению (нечувствительно к регистру) не только в именах пакетов, но и в их описаниях. В результате в выводе будет список всех пакетов, имеющих хоть какое-то отношение к искомому.

Для выделенного пакета доступно его краткое описание и формальные данные – принадлежность к группе, лицензия, объем подлежащего скачиванию архива и репозиторий, из которого будет получен пакет.

Более подробную информацию о пакете можно получить через меню Selection [Выделение]. Так, пункт Get file lists [Списки файлов] выведет список файлов и путей к ним в том виде, в котором они будут установлены в системе. Пункт Depends on [Зависит от] даст список зависимостей пакета, а Required by [Требуется] – список пакетов, которые зависят от выбранного.

Для установки найденного пакета достаточно пометить его и нажать кнопку Apply [Применить]. После этого некоторое время будут проверяться зависимости пакета, список которых выводится в специальной панели.

Нажатие кнопки Install [Установить] повлечет за собой скачивание пакета вместе со всеми его зависимостями, их распаковку и инсталляцию. Кнопка Cancel [Отмена] вызовет отказ от установки не только зависимостей, но и выбранного пакета.

Если все идет как надо, после описанных выше манипуляций мы будем иметь в системе установленный работоспособный пакет. Что и предлагается проверить в панели сообщения об успехе инсталляции – на ней имеется кнопка Run [Запустить], которая вызывает старт свежеустановленной программы.

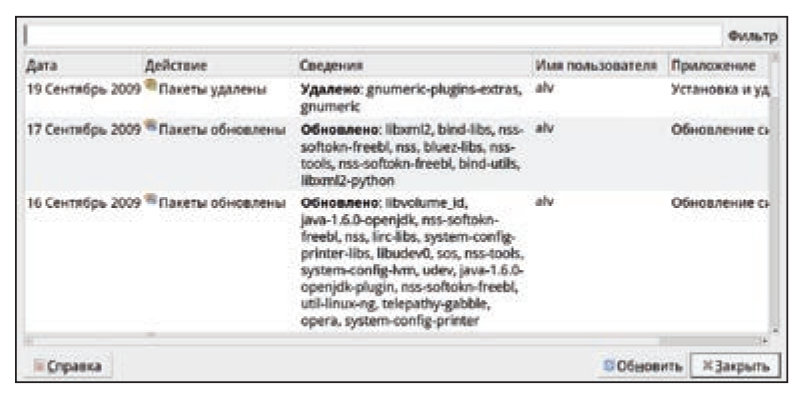
Однако нельзя исключить ситуации, что в ходе проверки зависимостей будут выявлены ошибки – как правило, они связаны с конфликтом версий пакетов, от которых зависит устанавливаемый. И единственное, что тут можно сделать – открыть вывод More details [Подробности], просмотреть его и закрыть панель ошибок. Выбранный пакет при этом, разумеется, установлен не будет.

Удаление пакетов происходит аналогично, только в обратном порядке: сначала снимается отметка с установленного пакета, затем нажимается кнопка Apply [Применить] – и наступает ожидание проверки зависимостей, завершающееся появлением окна со списком пакетов, которые будут удалены вместе с заказанным. Список очень внимательно изучается, после чего следует согласие на удаление или отказ от него.

Подчеркну необходимость *очень* внимательного изучения списка удаляемых зависимостей: они могут оказаться весьма неожиданными. Так, удаление пакета, установленного не индивидуально, а в составе какой-либо группы или коллекции (особенно при инсталляции), может нечаянно повлечь за собой снос половины системы.

PackageKit в Fedora 12 получит (за счет отдельных расширений) такие дополнительные возможности, как автоматическая установка пакетов по щелчку на имени файла в браузере или из командной строки – в ответ на сообщение «command not found».

Все действия по установке и удалению пакетов через *PackageKit* фиксируются в специальном файле – **/var/log/yum.log**; как явствует из названия, он не специфичен для *PackageKit*, а отражает действия через менеджер пакетов *Yum*. Однако *gnome-packagekit* предоставляет удобную форму визуализации его содержимого, вызываемую через пункты меню System > Software log [Система > Журнал установки], где показываются: дата действия и его характер (установка, обновление или удаление), имя совершившего его пользователя и приложения (субпакета в составе *gnome-packagekit*).



› Журнал установки, обновления и удаления пакетов.

По-хорошему, прежде чем заниматься установкой или удалением пакетов, неплохо бы выполнить некоторые подготовительные действия.

Во-первых, надо проверить доступные репозитории (те самые, которые подключались на стадии установки), что делается через меню System > Software sources [Система > Источники программ]. Скорее всего, все нужные источники пакетов из числа официальных для Fedora вообще и Russian Fedora в частности уже включены, но лишний раз убедиться в этом не мешает.

Затем имеет смысл обновить систему – через пункт меню System > Refresh package lists [Система > Обновление списка пакетов], который сначала приведет список доступных пакетов в актуальное (и соответствующее подключенным репозиториям) состояние, а затем предложит список пакетов, могущих быть обновленными, с которым остается только согласиться.

> «PackageKit — простое и удобное средство управления пакетами.»

И теперь обновление будет выполнено, если не произойдет ошибки – хотя нельзя исключить и последнего варианта. В этом случае придется обратиться к командной строке и *Yum*.

Пункты меню Software sources [Источники программ] и Refresh package lists [Обновление списка пакетов] вызывают самостоятельные субпакеты, входящие в *gnome-packagekit* – *gpk-repo* и *gpk-update-viewer*, соответственно. Но они могут быть запущены автономно, через главное стартовое меню среды – Система > Администрирование > Источники программ/Обновление программ.

Резюме

Из сказанного можно сделать вывод, что *PackageKit* в своей графической ипостаси – простое и удобное в обращении средство управления пакетами, функционально сходное с *Synaptic* для Deb-пакетов. В сравнении с последним он производит впечатление более медлительного. Однако это связано не с ним самим, а с RPM-форматом и базами данных для *Yum*, требующими скачивания существенно большего объема метаинформации. Второй недостаток *PackageKit* – трудность определения причин возникновения ошибок как при установке конкретного пакета, так и при тотальном обновлении системы. Это я отнес бы к некоторой недоработанности системы *PackageKit* в целом – ведь по сравнению с *Synaptic* она еще очень молода.

Однако и в своем нынешнем виде *PackageKit* пригоден для повседневного использования в сфере управления пакетами, если не выходить за пределы штатных ситуаций – а при их возникновении *Yum* нам в руки. LXF

Что за штука... Telepathy?

Марко Фиоретти открыл, каким образом части вашего – а заодно и чужого – рабочего стола могут взаимодействовать в режиме реального времени.

Что такое Telepathy?
Это не передача мыслей на расстояние, а штука, которая, возможно, уже работает внутри вашей Linux-машины: вы о ней и не подозреваете, но горько пожалеете, если она выйдет из строя.

А для чего нужна эта технология? Как она действует?
Telepathy (http://telepathy.freedesktop.org) – это каркас, «комплекс программных библиотек и протоколов» для унификации многих видов онлайн-коммуникаций в реальном времени.

Типа обмена мгновенными сообщениями (IM)?
Это самый частый вариант использования Telepathy, но потенциальная сфера его применения гораздо шире (если ее «освоит» достаточное количество Linux-приложений). Новая технология пригодится для организации голосовых переговоров через Интернет (VoIP), видеочатов и прямых соединений между различными клиентами протоколов реального времени.

> «Даже проект OLPC применяет Telepathy для коллективизации.»

Какую пользу может принести Telepathy при обмене мгновенными сообщениями?
Если в двух словах, то большинство современных клиентов недостаточно тесно увязаны с другими программами. Добавляя их поддержку в свои приложения, разработчики вынуждены ограничиваться единственным клиентом, или же дублировать значительную часть кода.

А зачем обычным настольным клиентам столь тесное взаимодействие?
Традиционный обмен мгновенными сообщениями заключается в том, что два пользователя пересылают друг другу небольшие кусочки текста в реальном времени, непосредственно между двумя компьютерами...

Ну да, знаю. И что здесь плохого?
Да ничего, но ведь общение ограничено экранной перепиской. На сегодня для программиста FOSS нет стандартного способа наладить такую же непосредственную связь в режиме реального времени между программами других типов.

Верно, но что даст на практике связь другого рода?
Простейший пример – коллективное редактирование и рецензирование, полностью конфиденциальное, быстрое и без обращения к посредникам (вроде Google Docs). Сейчас, когда вы с коллегой совместно работаете над текстовым документом, вам приходится не по разу обмениваться его разными версиями через свои клиенты IM, так?

Так. А если у наших текстовых процессоров будет Telepathy...
То они будут обмениваться данными прямиком между собой или через ваши IM-клиенты, используя те же самые учетные записи и настройки. Все, что вы набираете в своем процессоре, мгновенно (с учетом латентности сети, конечно) будет отображаться в текстовом процессоре коллеги. Уверен, вы уже представили себе выгоды от такого усовершенствования.

Да, я начинаю улавливать идею. Но ведь Telepathy не ограничивается работой с текстом?
Ни в коем случае! Теоретически, технология Telepathy может в той или иной форме использоваться в любой части рабочего стола. Например, с помощью Telepathy можно играть в компьютерные игры на расстоянии, открывать удаленным пользователям доступ к папкам на своем винчестере или обмениваться музыкальными плей-листами в реальном времени. Возможно также распространение информации о присутствии для программ разного рода (например, плазмоидов KDE 4).

Что такое информация о присутствии?
Это уведомление других интернет-пользователей о том, что вы находитесь в сети и готовы к общению. Естественно, кому и когда можно сообщать такую информацию, контролируете вы. Среди прочего, Telepathy обеспечивает стандартизацию и этой функции.

Готова ли технология Telepathy к использованию?
Да, и уже используется во многих приложениях: например, Gnome задействует Telepathy в своей программе обмена сообщениями *Empathy*. Одно из применений Telepathy – обеспечение общего доступа к файлам и папкам для абонентов *Empathy* через инструмент удаленного доступа к рабочему столу *Vino*. Кроме того, возможна отправка файлов тем же абонентам с помощью модуля *nautilus-sendto* в составе *Nautilus*. А есть еще проект OLPC, где Telepathy предполагается использовать для определения присутствия и коллективной работы. Несколько текущих проектов направлено на реализацию совместной работы через Telepathy, сбор информации о присутствии и тому подобных сервисов в *Plasma*, интерфейсе KDE. Многие свободные настольные программы уже примут на вооружение Telepathy к моменту выхода данной статьи.

Текстовые процессоры в том числе?
Да. Существуют модули коллективной работы для *AbiWord* (*AbiCollab*, www.abisource.com/wiki/AbiCollab) и текстовых редакторов, например *Gedit* (http://alban.apinc.org/blog/collaborative-editing).

А как насчет обмена музыкой и игр, о котором вы упомянули?
Модули для *Rhythmbox* и *Banshee* позволяют изменить онлайн-статус в зави-



симости от воспроизводимой дорожки, а также непосредственно обмениваться музыкальными файлами с абонентами IM. По части игр – благодаря Telepathy, можно играть в многопользовательском режиме в *Sudoku*, *Tetris* (через *GTetrinet*, http://gtetrinet.sourceforge.net)и*PyHalebarde* (https://launchpad.net/pyhalebarde).

Теперь о технических деталях. Вы сказали, цель технологии — унифицированное управление связью в реальном времени. Почему же тогда Telepathy называют не программой или демоном, а каркасом?
Это один из постулатов Unix и Linux: «делай одно дело, и делай его хорошо». Telepathy устанавливает правила игры и обеспечивает основной «инвентарь» для нее: это комплект библиотек с единым интерфейсом и стандартизация служб различного назначения.

И это хорошо: программистам теперь меньше работы!
Конечно! Если есть общая система связи, то для добавления новой функции достаточно написать небольшой модуль. Кроме того, можно смешивать ПО с разными лицензиями, а также увеличивать производительность за счет загрузки в память только тех компонентов, которые непосредственно используются приложениями.

Такая архитектура более безопасна и надежна, правда?
Вы правы: зависание одного компонента не приведет к краху всего комплекса. Приложение состоит из отдельных модулей, и к каждому из них можно применить свой вариант политики безопасности SELinux.

А как это делается? По волшебству?
Telepathy основывается на шине сообщений D-Bus, которая принята в Gnome и KDE 4 для обмена информацией между процессами.

Позвольте, но ведь D-Bus годится только для связи между приложениями, работающими на одном компьютере?
Верно, но Telpathy предлагает механизм для обмена произвольными сообщениями, в том числе, D-Bus по сети, называемый «трубами» [tubes]. Трубы, «проложенные» между двумя или несколькими пользователями, объединяют объекты Telepathy, работающие на разных компьютерах.

Каково внутреннее устройство Telepathy?
Основные компоненты – менеджеры соединений и канальные объекты, плюс компонент Mission Control. Первая составляющая – объекты, которые обеспечивают соединение с учетной записью IM или VoIP и предоставляют интерфейсы для отправки или получения информации о присутствии, управление псевдонимами и пр. Один менеджер соединений может управлять несколькими протоколами и соединениями.

А что такое канальные объекты?
Это списки контактов или настроек учетных записей с интерфейсами для чтения, установки различных параметров и обмена данными.

Наконец, самая важная часть?
Mission Control – единственный компонент Telepathy, который должен дежурить постоянно. Это центр управления, который вовремя запускает менеджеры соединений и следит за созданием канальных объектов и управлением ими. Кроме того, с помощью Mission Control пользователь уведомляет других людей о своем присутствии (или скрывает его).

Где можно узнать о разработке ПО, совместимого с Telepathy?
Для начала изучите руководство разработчика Telepathy (*Telepathy Developer's Manual*) (http://people.collabora.co.uk/~davyd/telepathy-book) и взгляните на примеры использования Telepathy API на http://telepathy.freedesktop.org/wiki/recipes. Если хотите программировать для KDE, посетите http://techbase.kde.org/projects/telepathy.

Сильно ли повлияет Telepathy на развитие свободного ПО?
Если Telepathy ограничат традиционными рамками клиентов IM и VoIP, это будет несправедливой недооценкой новой технологии. По-настоящему значительное влияние Telepathy окажет в том случае, если ее поддержку включат во многие менеджеры рабочих столов и приложения. Если это произойдет, то новая технология может стать важным компонентом социального рабочего стола, о котором мы говорили в LXF122. LXF



По рецептам доктора Брауна

Д-р Крис Браун
Доктор обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

Добротное администрирование систем из причудливых заворотов кишок серверной.

Открыто всегда

Говоря «открытый», мы обычно думаем об открытом ПО, но несколько разговоров на недавней конференции UK Unix and Open System User Group (UKUUG) в Бирмингеме напомнили мне, что этим словом называют не только ПО, которое сражается с проприетарным. Первый был с Томом Йейтсом [Tom Yates] из Gatekeeper Technology, описавшим Openmoko как «почти полностью открытый телефон GSM»: на нем установлен набор ПО Linux, но можно получить и полную электрическую схему и даже CAD-файлы для изготовления корпуса. Телефон остается аппаратом для энтузиастов и еще не готов к выпуску, о чем Том честно мне поведал: «Нет, он еще не работает "из коробки"». И даже: «Кто привык к iPhone, тот возненавидит этот прибор». Толпа поклонников iPhone, пожалуй, и правда не готова к Openmoko, но если у вас хакерский зуд, зайдите на wiki.openmoko.org.

Наносим будущее на карту

Шаран Муни [Ciaran Mooney] рассказывал о проекте Open Street Maps. Это открытые цифровые карты. Идея состоит в том, чтобы объединить тысячи добровольцев по всему миру, которые гуляют, ездят на велосипедах или на машинах по улицам с GPS-приемником в руках и загружают свои трассы в открытую базу. Они добавляют тэги с названиями дорог, типами поверхности – дорога, тропинка, велосипедная дорожка, железная дорога, канава и т.д. – и интересные точки: маяки, поля для гольфа и велосипедные стойки. Проекту может содействовать любой – программистом быть не обязательно! Чтобы увидеть результаты или принять участие, зайдите на сайт www.openstreetmap.org.

Мне также понравилось и руководство по Arduino от Эндрю Элиаша [Andrew Eliasz], и я провел счастливый день, втыкая резисторы и светодиоды в макетную плату, имитируя светофор (на C) и проигрывая мелодию «*Три слепые мыши*» (в си бемоль) на пьезоэлектрическом элементе. Чистая радость...













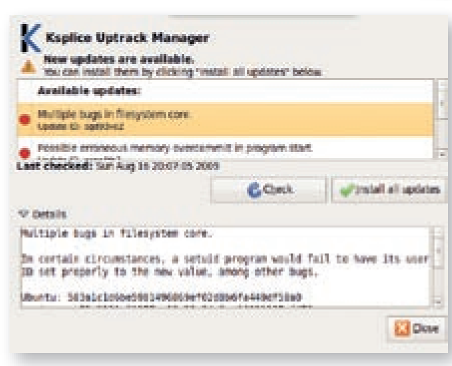



Снизим простои

Ksplice Держите серверы при деле, сэкономив на обновлениях.

LINUX FORMAT На DVD
» Ksplice 0.9.9

Есть одна история об интернет-провайдере, которую я люблю рассказывать студентам. Пользователи обнаружили, что версия ядра Linux на его серверах – трехлетней давности. На вопрос «почему» был дан следующий ответ: «А когда мы последний раз загружали компьютер, это была самая свежая версия». Эта притча была призвана подчеркнуть стабильность Linux, но у нее есть и обратная сторона. Почему этот нерадивый провайдер не устанавливал заплатки ядра с обновлениями безопасности? Потому что обновление ядра обязательно потребует перезагрузиться?



› На взгляд с прищуром *Ksplice Uptrack Manager* напоминает *Ubuntu Update Manager*.

Больше никогда?

А вот и нет, если верить ребятам с www.ksplice.com. Они предоставляют сервис Ksplice Uptrack, применяющий заплатки к работающему ядру без перезагрузки. Сначала я счел было это маркетинговым ходом, но, копнув глубже, случайно нашел отчет www.ksplice.com/paper, убедивший меня, что за этим сервисом стоит реальная технология. Она предоставляется в виде deb-пакета, несложного в установке. Пока скачать его бесплатно можно только для Ubuntu 9.04 и 9.10. В обмен на потраченное место на диске, на панели задач появится иконка *Ksplice Update Manager*, и когда становятся доступны новые обновления, на ней возникает красный треугольник. Щелкнув по ней, вы попадете в окно, напоминающее *Ubuntu Update Manager*. Если Ksplice окажется достаточно интересным, сервис появится и в других дистрибутивах, и, возможно, мы даже не прочь будем за него заплатить.

Когда перезагрузка?

Чтобы узнать, как часто и зачем перезагружаются серверы, мы провели опрос на английском сайте **LXF**. Для 30 респондентов, которые в общей сложности управляли 190 серверами, средний интервал между перезагрузками составил 159 дней. Наиболее частой причиной перезагрузки было обновление ядра – об этом сообщили 19 из 30 респондентов; похоже, это плодородная почва для разработчиков Ksplice. Между тем, только двое сообщили, что выключали серверы для тестовых запусков и чисток, при сбоях электричества и обесточивании. Наконец, один пользователь перезагружал сервер для применения обновлений прошивок, а другой – для изменений на уровне ПО.

Списки управления доступом

ACL Только доктор решил, что одолел *chmod*, как нашелся ACL.

В девяностых годах прошлого века комитет POSIX по стандартизации (IEEE 1003.1e) работал над набором расширений безопасности для Unix. Когда финансирование прекратилось, от стандарта отказались, но во многом он уже был завершен, и кое-что из него перекочевало в Linux, например, списки управления доступом (ACL).

Стандартная модель доступа в Linux – тропа хоженая: у нас есть права на чтение, запись и выполнение, и установить их можно для владельца файла, членов группы владельца и всех остальных. И это все. Списки управления доступом расширяют эту модель, так что права на чтение, запись и выполнение можно задать для любого количества пользователей и групп.

Файловые системы ext2 и ext3 поддерживают ACL, но только в том случае, когда они смонтированы с опцией **acl**. Например:

```
sudo mount -o acl /dev/sda2 /home
```

Конечно, более солидный подход – добавить параметры монтирования в **/etc/fstab**, например, так:

```
/dev/sda2 /home ext3 acl        0        0
```

Если ACL активированы, управлять ими можно с помощью утилит командной строки *setfacl* и *getfacl*. На рисунке справа показаны синтаксис команды **setfacl** и способ представления ACL.

Давайте выполним несколько команд, чтобы посмотреть, как все это работает. Сначала создадим файл **foo** и установим права доступа к нему в 600:

```
touch foo
chmod 600 foo
ls -l foo
-rw------- 1chris chris02009-08-1814:45foo
```

Обратите внимание, что в ACL файла автоматически добавятся три раздела, соответствующие обычным правам доступа (вывод команды также содержит строки **file**, **owner** и **group**, которые мы на данном уроке не меняем).

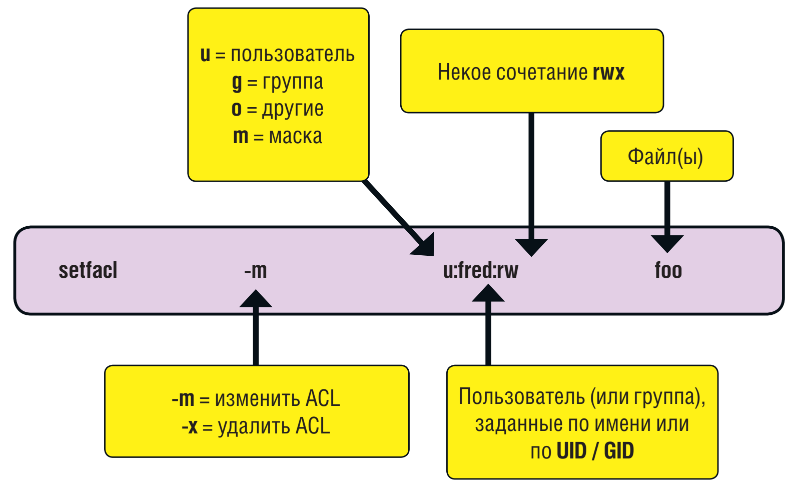
```
getfacl foo
user::rw-
group::---
other::---
```

На самом деле эти разделы автоматически синхронизируются с обычными правами доступа – изменения, произведенные командой **chmod**, появятся в выводе **getfacl**, а изменения **setfacl** появятся в выводе **ls -l**.

Создадим еще два ACL, добавив доступ на чтение-запись для пользователя Tom и доступ на чтение для группы Games:

```
setfacl -m u:tom:rw foo
setfacl -m g:games:r foo
getfacl foo
user::rw-
user:tom:rw-
group::---
group:games:r--
mask::rw-
other::---
```

Дело немного усложнится, если задать маску ACL. Она используется для ограничения доступа явно поименованных в ACL пользователей и групп и не затрагивает права владельца файла. Здесь мы устанавливаем маску только для чтения и снова просматриваем ACL. Обратите внимание, что фактические права Тома были ограничены маской; теперь это только чтение, а не чтение и запись:



Анатомия команды *setfacl*.

```
setfacl -m m::r foo
getfacl foo
user::rw-
user:tom:rw-        #effective:r--
group::---
group:games:r--
mask::r--
other::---
```

Наконец, можно удалить ACL таким образом:

```
setfacl -x g:games: foo
getfacl foo
user::rw-
user:tom:rw-
group::---
mask::rw-
other::---
```

Команда **ls** не пытается отобразить полный список управления доступом для файла, но позволяет понять, что он задан (об этом говорит знак **+** после девяти бит прав доступа):

```
ls -l foo
-rw-r-----+ 1 chris chris02009-08-1814:45foo
```

Задать список управления доступом для файла можно и по-другому. Во-первых, можно заставить **setfacl** прочитать ACL из файла или из стандартного ввода. Следующий пример показывает, как скопировать ACL с **foo** в **bar**:

```
getfacl foo1 | setfacl --set-file=- bar
```

Во-вторых, можно заставить вновь созданные файлы унаследовать ACL от содержащего их каталога. Чтобы сделать это, нужно назначить каталогу список управления доступом по умолчанию. Вот пример. Сначала создадим каталог acltest и назначим ему ACL по умолчанию (опция **-d** команды **setfacl**):

```
mkdir acltest
setfacl -d -m u:tom:rw,g:games:r acltest
```

Если теперь создать новый файл в этом каталоге и просмотреть его ACL, мы увидим списки для Tom и Games, которые пришли от списка по умолчанию каталога **acltest**.

Исходя из моего опыта, ACL используются не слишком часто. Большинство пользователей не представляют себе всех возможных сценариев, которых можно достичь стандартными правами доступа, осторожно используя группы и, возможно, установкой битов **setuid** и **setgid** (рассмотрены на следующей странице).

Сказание о трех битах

Биты режима Прослывите на работе матерым хакером, обретя знания о setuid, setgid и загадочном sticky bit.

Большинство людей знает о девяти битах, определяющих права доступа к файлу – rwxrwxrwx – но есть еще три, которые не столь известны. В этом месяце я решил вывести их на сцену. Это (слева направо):

» Бит установки идентификатора пользователя; друзья зовут его setuid.

» Бит установки идентификатора группы; сокращенно setgid.

» «Липкий» бит (sticky bit).

Установить и сбросить эти биты, как и остальные биты режима, можно с помощью **chmod**. Например, команды

```
$ chmod u+s foo
$ chmod g+s foo
$ chmod +t foo
```

включат setuid, setgid и sticky bit соответственно.

Если вы предпочитаете задавать режимы доступа восьмеричными числами, знайте, что setuid имеет значение 4000, setgid – 2000, а sticky bit – 1000. Таким образом, команда

```
$ chmod 4755 foo
```

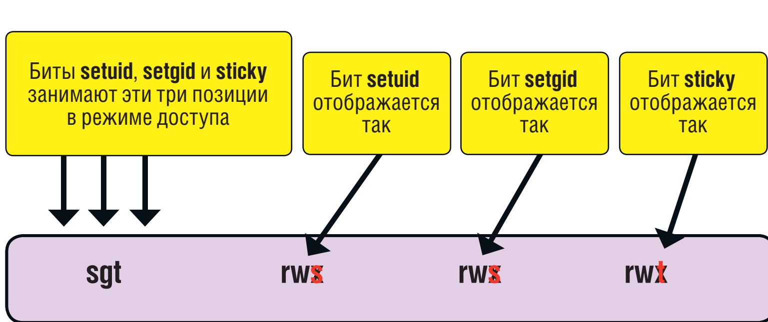
сделает файл **foo** исполняемым для всех и включит бит setuid.

Понимание вывода команды ls -l

В выводе команды **ls -l** эти биты представлены довольно необычно. Казалось бы, проще всего было добавить три символа в начало строки. Вместо этого три символа прав на исполнение замещаются, как показано на рисунке ниже. Таким образом, бит setuid отображается как 's' на месте бита прав на исполнение для владельца файла, setgid – на месте бита прав на исполнение для группы, а sticky bit отображается как 't' на месте бита прав на исполнение для всех остальных.

Следующие команды помогут разобраться в этом:

```
$ls -l foo
-rwxr-xr-x 1 chris chris 833592 2009-08-16 11:07 foo
$ chmod u+s foo # Установим setuid
$ls- l foo
-rwsr-xr-x 1 chris chris 833592 2009-08-16 11:07 foo
$ chmod g+s foo # Установим setgid
$ls -l foo
-rwsr-sr-x 1 chris chris 833592 2009-08-16 11:07 foo
$ chmod +t foo # Установим sticky bit
$ls -l foo
-rwsr-sr-t 1 chris chris 833592 2009-08 -16 11:07 foo
$ chmod 4755 foo # setuid bit, using octal
$ls -l foo
-rwsr-xr-x 1 chris chris 833592 2009-08-16 11:07 foo
```



› На рисунке показано, как биты выполнения перезаписываются, чтобы показать наши три бита.

> **Битком набито**
>
> Файлов, у которых setuid, setgid и sticky bit установлены, относительно немного, и встретишь их нечасто. Но их можно легко найти с помощью опции **-perm** команды **find**.
>
> Следующая команда найдет все файлы, у которых установлен setuid на пользователя root:
>
> ```
> $ sudo find / -perm /4000 -user root
> ```
>
> следующая – все каталоги, у которых установлен setgid:
>
> ```
> sudo find / -type d -perm /2000
> ```
>
> следующая – все каталоги с установленным sticky bit:
>
> ```
> sudo find / -type d -perm /1000
> ```
>
> Проверьте также, есть ли хотя бы один файл со sticky bit (скорее всего, таких нет):
>
> ```
> sudo find / -type f -perm /1000
> ```

И еще одна штука, о которой знают немногие. Так как 's' или 't' вытесняют 'x', как понять, установлено для файла право на исполнение или нет? Ответ: если оно не установлено, вы увидите заглавные 'S' или 'T' вместо строчных. Например:

```
$ chmod 4644 foo
$ls -l foo
-rwSr--r -- 1 chris chris 833592 2009-08-16 10:51 foo
```

Такого быть не должно. Если программа работает в режиме setuid, она обязана быть исполняемой. Вот крайний случай:

```
$ chmod 7000 foo
$ls -l foo
---S--S--T 1 chris chris 833592 2009-08-16 11:07 foo
```

Я смог придумать только одно применение этого примера: показать нахальному сисадмину с вашей работы, что вы еще покруче него.

Setuid сияет

Теперь, разобравшись, как задать и отобразить эти расширенные права, изучим, что они, собственно, делают. А попутно рассмотрим случаи, когда эти права вам пригодятся.

Гвоздь программы – несомненно, setuid. Если установить его исполняемому файлу, он повысит действующий идентификатор пользователя процесса, выполняющего файл, до владельца программы, которым обычно является root. Обратите внимание, что у каждого процесса есть две «личности»: реальная и действующая. При принятии решений, связанных с предоставлением процессу доступа к ресурсам, используется действующая. Если бит setuid не установлен, реальная и действующая «личности» одинаковы. Значение этой простой, но элегантной возможности в Linux трудно переоценить, потому что она лежит в сердце всех утилит, отвечающих за повышение привилегий, таких как **sudo**.

Найти применение для setuid нетрудно. Предположим, вы написали игру (назовем ее *Spacewar*, хотя я боюсь, что это название уже украли в 1961 году) и хотели бы работать с файлом, содержащим таблицу из 10 рекордных результатов игроков. Конечно, просто открыть его на запись для всех нельзя – игроки сжульничают; но программа должна иметь возможность сохранять там очки от имени игрока, если он решит, что результат того достоин. Итак, вы создаете новую учетную запись пользователя; назовем его spaceman. Вы запрещаете непосредственный вход в систему

с этой записи – она существует только затем, чтобы дать владельца файлу турнирной таблицы. Теперь создайте файл таблицы, позаботившись, чтобы ее владельцем стал spaceman и право на запись было только у него. Сделать это можно примерно так:

```
# cd /var/run
# mkdir spacewar
# cd spacewar
# touch league-table
# chown spaceman league-table
# chmod 644 league-table
```

Создастся файл, который смогут читать все, но изменять – только spaceman.

Вторая часть трюка – сделать так, чтобы программа запускалась с setuid на пользователя spaceman таким образом:

```
# cd /usr/local/bin
# chown spaceman spacewar
# chmod 4755 spacewar
```

Теперь запускать *spacewar* смогут все. При этом она выполнится с эффективными правами пользователя spaceman, а значит, сможет писать в файл турнирной таблицы.

Углубляемся в setuid

Классический пример программы с setuid – *passwd*, утилита, позволяющая пользователям менять пароли путем записи хэша нового пароля в **/etc/shadow**. Посмотрите внимательно на владельцев и права доступа в следующем примере:

```
$ ls -l /usr/bin/passwd /etc/shadow
-rw-r----- 1 root shadow 1262 2009-08-04 16:16 /etc/shadow
-rwsr-xr-x 1 root root 42776 2009-04-04 06:50 /usr/bin/passwd
```

Программа смены пароля имеет права на выполнение для всех и запускается с setuid на пользователя root, что дает ей возможность записывать данные в файл shadow.

Программы, у которых установлен setuid, особенно если их владельцем является root, должны быть надежными и устойчивыми к попыткам заставить их делать вещи, не входившие в замысел автора. Принцип наименьших привилегий предполагает, что если нужно поднять привилегии, то лучше делать это не с пользователем root, как в нашем примере со spaceman. С этими словами я насчитал в RHEL5 34 программы с установленным setuid, и владельцем каждой был root!

> «Нужно было назвать его битом ограниченного удаления.»

Бит setuid представляет собой реальную угрозу безопасности, когда используется в сочетании со сменными дисками. Это происходит так: Мистер Нехороший Человек на домашнем компьютере записывает на CD копию *Bash* с владельцем root и включенным setuid. Потом приносит диск на работу, вставляет его в сервер, ждет, пока он автоматически смонтируется – и вот он в консоли root. Как пресечь его злодеяния? Ну, по умолчанию только у пользователя root есть право монтировать файловые системы. Чтобы остальные пользователи могли монтировать сменные диски, в файле **/etc/fstab** есть запись с опцией монтирования **user**. Эта опция подразумевает **noexec**, **nosuid** и **nodev**; среди прочего, это означает что бит setuid не будет учитываться ни в одном файле в этих файловых системах.

За дело берется setgid

Бит setgid в применении к исполняемому файлу похож на setuid, но он устанавливает эффективный идентификатор группы процесса в идентификатор группы исполняемого файла. Это другой способ повышения привилегий. Мне сложнее найти пример его эффективного использования, хотя, например, он применяется **postdrop** и **postqueue**, компонентами почтовой системы *Postfix*. Но мне всегда казалось, что setgid находится немного в стороне по сравнению с setuid, как этакий предпоследний из могикан. Однако в дополнение к основной работе у setgid есть еще одна функция, применяемая к каталогам: он изменяет правила получения группы владельца для файлов, создаваемых в каталоге. Обычное правило, конечно, таково: группа создаваемого файла есть основная группа его создателя. Если установлен бит setgid, группа файла наследуется от группы каталога. Это удобно в некоторых интересных случаях разделения групп.

Бит	Восьмеричное значение	Значение для файла	Значение для каталога
Setuid	4000	Файл выполняется с действующим UID владельца	Игнорируется
Setgid	2000	Файл выполняется с действующим GID группы	Файлы, создаваемые в каталоге, наследуют группу от группы каталога
Sticky	1000	Игнорируется	Файлы в каталоге могут быть удалены только их владельцем

› Выше вы найдете краткое описание каждого бита и его применение.

Sticky bit возвращается

Во времена юности Unix sticky bit означал: сохранить образ этого исполняемого файла на устройстве подкачки после того, как его выполнение закончено, потому что есть вероятность, что скоро он понадобится снова. Он мог бы использоваться в программах вроде оболочки, редактора и компилятора C. Более быстрые диски и виртуальная память с подкачкой страниц по требованию сделали этот бит ненужным, и его карьера пошла на спад. На долгие годы о нем забыли. Однако недавно он вернулся в Linux в совершенно новом качестве. Применительно к каталогу он изменяет правила, касающиеся того, кто может удалять файлы в этом каталоге. Без sticky bit все пользователи, у которых есть права на запись в каталог, могут удалять файлы в нем (даже если они не являются их владельцами). Установка «липкого» бита меняет правила – с ним можно удалять только те файлы, владельцем которых вы являетесь. Вообще нужно было назвать его битом ограниченного удаления, но sticky bit к нему, э-э... прилипло. Классический пример его использования – временные каталоги, вроде **/tmp**. В отношении обычных файлов Linux этот бит ничего не меняет. LXF

Патент на setuid

Деннис Ритчи [Dennis Ritchie], изобретатель механизма setuid, запатентовал его в 1973 году (Патент США 4135240). Патент описывал бит через логическую схему, потому что тогда никто не был уверен, что программные реализации можно патентовать. Патент содержал фразу: «Функционально эквивалентная программная реализация признается частью описанного здесь вклада изобретателя. Для определенных целей программная реализация может быть более предпочтительной». Кстати, слово «предпочтительной» содержало ошибку. Патент можно найти на сайте patft.uspto.gov. Его стоит прочесть, пожалуй, только как образец искусства запутывания юристов-патентоведов. Один из самых светлых пассажей гласит: «Посредством чего вышеуказанный текущий пользователь может получить выборочный доступ к файлам вышеупомянутой компьютерной системы, владельцем которых является указанный пользователь, во время исполнения указанных команд программы». Ну, как?

Дуг Макплрой [Doug McIlroy], бывший глава исследовательского отдела Bell Labs, в котором родился Unix, прокомментировал: «AT&T распространяла Unix, понимая, что с пользователей можно будет собирать лицензионные платежи, когда будет выдан патент на setuid. Когда это, наконец, произошло, логистические проблемы сбора мелких лицензионных платежей с сотен лицензий показались не стоящими усилий, и патент был передан в общественное достояние». И это удача, потому что воплотить Linux таким, как он есть сейчас, без постоянного нарушения этого патента было бы невозможно.

Решите проблемы **лицензирования ПО** с помощью профессионалов!

Операционная система GNU/Linux и свободное программное обеспечение помогут вам с минимальными затратами решить проблему лицензирования программного обеспечения, повысить безопасность и надежность вашей компьютерной сети.

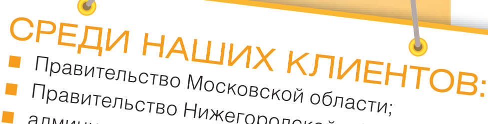
Компания ГНУ/Линуксцентр предлагает вам внедрение ОС GNU/Linux и свободного программного обеспечения, реализацию и техническую поддержку сложных технических решений на базе свободного ПО, обучение ваших сотрудников — как пользователей, так и технических специалистов.

С НАШЕЙ ПОМОЩЬЮ ВЫ СМОЖЕТЕ:

- оптимизировать затраты на лицензирование ПО за счет максимально возможного использования свободного ПО;
- существенно сократить время системных администраторов, затрачиваемое на устранение последствий деятельности вирусов и сбоев в программном обеспечении.

ТИПОВЫЕ ПРОЕКТЫ:

- миграция рабочих станций и серверов с Microsoft Windows на GNU/Linux;
- установка 1С на серверах и рабочих станциях под управлением GNU/Linux;
- миграция с Microsoft Windows Active Directory на Mandriva Directory Server;
- миграция с Microsoft Exchange на Zimbra;
- внедрение интернет-телефонии на базе Asterisk;
- внедрение свободной CRM-системы SugarCRM;
- создание кластеров высокой доступности;
- реализация терминальных решений;
- создание порталов любой сложности на базе свободных CMS-систем — Joomla, Drupal, Plone;
- внедрение защищенных систем на основе сертифицированного ФСТЭК ПО.



СРЕДИ НАШИХ КЛИЕНТОВ:

- Правительство Московской области;
- Правительство Нижегородской области;
- администрация Черниговского района Приморского края;
- Министерство финансов республики Саха (Якутия);
- Владивостокский государственный университет экономики и сервиса;
- группа компаний «ИМАГ»;
- компания «Азбука мебели»;
- компания «Бестли — выставочные материалы»

и другие организации различного профиля.

Наш опыт внедрения свободного программного обеспечения в компаниях различного профиля поможет выбрать оптимальное сочетание свободного и коммерческого программного обеспечения, подходящее именно для вашей организации, а также поможет избежать техничеких и организационных проблем при внедрении свободного ПО.



Департамент внедрений компании ГНУ/Линуксцентр
Телефон в Москве: (499) 271-49-54,
в Санкт-Петербурге: (812) 309-06-86



Учебники

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Новый «ассемблер» для ОС?

Нам нет необходимости наступать на те же грабли, что уже были.
В.С. Черномырдин

Пару лет назад, утверждая, что никакая здравая коммерческая структура не будет заниматься фундаментальными исследованиями, я всегда упоминал о двух исключениях из этого правила: Bell Labs и IBM.

К сожалению, на сегодня осталось только одно исключение, но сила умершей для фундаментальной науки Bell Labs была в людях, а эти люди «зажигают» и сейчас.

О ком это я? А я о Робе Пайке [Rob Pike] и Кене Томпсоне [Ken Thompson] (Роберт Гризмер [Robert Griesemer] тоже молодец, но создатели Unix все-таки известнее). Они опять изобрели язык Си и назвали его Go (спасибо Google). Старая гвардия почистила изначальный Си, а также добавила туда сборщик мусора и возможность распараллеливания вычислений. Эдакая эволюция.

Интересно, сможет ли Google Go стать новым «ассемблером» для операционных систем и наследником Си или сделает «финт ушами» и со временем добьет отчаянно не сдающийся Fortran? Простота Си и параллельные вычисления выглядят заманчиво. Поглядим, что будет лет через пять.

P.S. Вышел *TeX Live 2009*, то есть любопытным ТеХникам настала пора обновиться и, например, «потыкать палочкой» *luatex*.

E.m.Baldin@inp.nsk.su

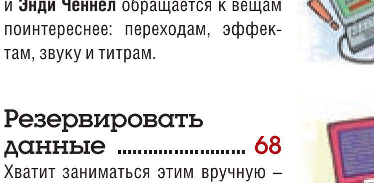
В этом месяце вы научитесь...



















Совет месяца: Cwiid



Допустим, вы собрались показать презентацию в *Impress*, но не хотите все время стоять у клавиатуры. Разумеется, тут выручает пульт дистанционного управления; но даже если он и заработает в Linux, всегда есть шанс потерять ту маленькую штучку, что втыкается в USB-порт. Но если вы – счастливый владелец Nintendo Wii (подумайте над этим), есть и другое решение, поскольку Wiimote соединяется с приставкой через Bluetooth.

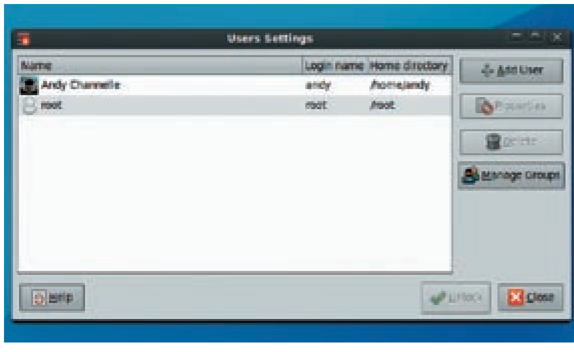
Cwiid (http://abstrakraft.org/cwiid) превращает нажатия на кнопки Wiimote в события клавиатуры и перемещения мыши. Вам понадобятся две программы: *wmgui*, выводящая информацию, полученную с Wiimote (для тестирования), и *wminput* – для реальной работы. Запустите их из терминала и нажмите на кнопки 1 и 2 на пульте по запросу; после этого кнопкам Wiimote будут назначены события мыши и клавиатуры, а покачивание пульта позволит перемещать курсор. Различные конфигурационные файлы в **/etc/cwiid/wminput** покажут, как настроить отображение органов управления, хотя в варианте по умолчанию «крест» уже соответствует курсорным клавишам, чего должно хватить для управления презентацией.

Если при старте *wminput* выдает сообщение «unable to open uinput» [не могу открыть устройство uinput], попробуйте запустить программу от имени root: права на устройства uinput обычно строги. Man-страница *wminput* объяснит, как заставить программу работать от имени обычного пользователя. Дополнения тоже поддерживаются – добавив Guitar Hero, вы сделаете свое выступление незабываемым...

Безопасность:

Защитить свои файлы от внешних угроз – дело хорошее, но **Энди Ченнел** нашел более серьезный фактор риска для ваших данных... ваша родня!





› **Перейдите в User Settings [Управление пользователями], чтобы добавить или удалить пользователей.**



Наш эксперт

Энди Ченнел
Энди, видимо, будет делать Первые шаги в Linux вечно, а технологиями он заинтересовался, открыв для себя Dragon 32.

Обычно говорят, что Linux имеет лучшую модель безопасности по сравнению с другими ОС, поскольку его подход «root/обычный пользователь» предотвращает случайное или намеренное повреждение системных файлов запросом пароля root (суперпользователя) при всех важных действиях. Это спасает вашу систему; но для большинства рядовых пользователей системные файлы ОС – наименее ценная часть компьютера. Данные (фотографии, учетные записи, контакты, электронные письма и т.д.) – вот что им важнее всего. Уязвимости ОС могут дать атакующему доступ к данным локально и удаленно, но мы-то поговорим о предотвращении неавторизованного или случайного доступа к вашим данным по более прозаичным причинам.

Linux – многопользовательская ОС. При установке дистрибутива вам предлагается создать одного или нескольких пользователей (это можно сделать и после), и каждый имеет личные настройки, рабочий стол и каталог для своих файлов в **/home/имя_пользователя**. Чаще всего это единственное место, где обычный пользователь может сохранять или записывать информацию – значит, здесь и будут храниться документы и фото; но не исключены и файлы, специфичные для приложений (например, пользовательские настройки), а то и приложения целиком.

Если во время установки вы недосоздали всех необходимых пользователей, поищите Управление пользователями [User Management] в разделе Настройки [Preferences]. В Ubuntu имеется набор опций для определения привилегий каждого пользователя, среди которых – запрет монтирования внешних устройств, использования принтера или сканера и подключения к Интернету. Однако пока мы с ними работать не будем.

Вместо этого откройте свой менеджер файлов и перейдите в каталог **/home**, где находятся директории, названные по именам всех пользователей, зарегистрированных в системе. Мы используем Gnome, но в KDE действия аналогичны. Можно применить метод «тотального запрета» и закрыть весь наш каталог, но это вряд ли будет продуктивно: в какой-то момент может понадобиться использовать файлы совместно с другими членами семьи.

Одни равнее других

Дважды кликните на вашем каталоге, чтобы увидеть свои файлы и директории. Выберите один из каталогов, щелкните на нем правой кнопкой мыши и выберите Свойства [Properties], затем перейдите на вкладку Права [Permissions]. По умолчанию каталог должен быть открыт для владельца на Чтение [Read] и Запись [Write] или Создание и удаление файлов [Create and Delete files], тогда как настройки Группы [Group] и Остальных [Others] должны предоставлять доступ только на чтение. В большинстве случаев этого достаточно, но что если вы планируете вечеринку-сюрприз для любимой(-ого) и хотите скрыть от ее (его) любопытных глаз все данные о приготовлениях? При стандартных настройках ваши документы можно будет прочесть (хотя и нельзя изменить).

Очередной ступенькой вверх по лестнице безопасности будет установка для Группы и Остальных значения Просмотр содержимого [List Files only]. Другие пользователи смогут лишь просматривать ваши каталоги, а любая попытка открыть документ вызовет сообщение «Мало прав» [insufficient privileges]. Этак уже лучше, но напоказ все еще выставлено немало метаданных (дата создания, размер файла), способных раскрыть ваши планы.

Следующая опция эквивалентна запиранию всех замков: установите для пункта Доступ к каталогу [Folder Access] значение Нет [None]. Тогда другой пользователь, попробовав влезть в ваш каталог, узнает, что у него на это не хватает прав. Правда, потом придется изобретать причину, с чего это у вас завелись секреты...

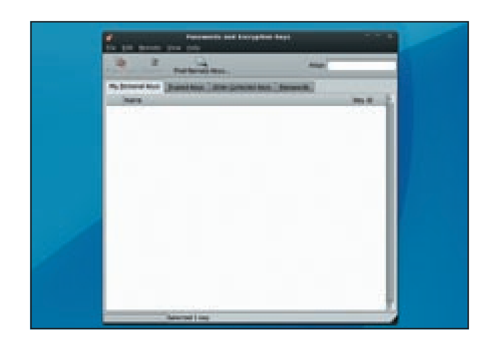
» **Месяц назад** Мы освоили горячие клавиши и расширили *Firefox*.

Рулим правами



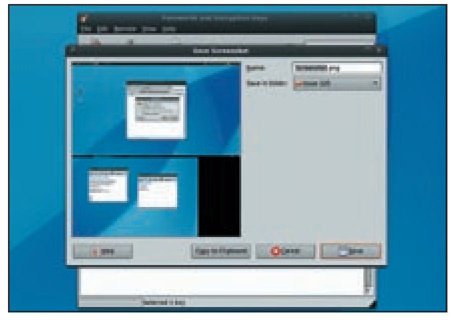
Шаг за шагом: Шифруем и дешифруем данные

Для вящей безопасности, зашифруем информацию в одном каталоге или во всей домашней директории. Проделаем это в Ubuntu, но процесс будет таким же и в других Gnome-дистрибутивах.



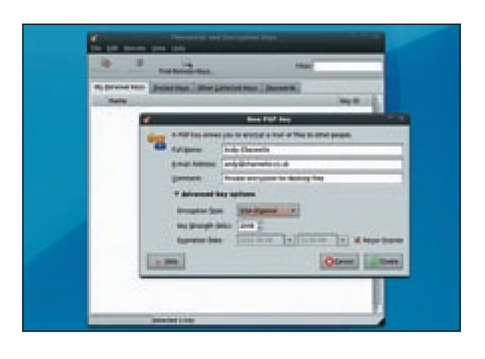
1 Создаем ключ шифрования

Чтобы зашифровать каталог целиком, щелкните на нем правой кнопкой мыши и выберите пункт Зашифровать [Encrypt]. Скорее всего, если вы ранее не выполняли шифрование, появится сообщение о том, что необходимо сформировать ключ. К счастью, тут же откроется диалог, в котором вы сможете начать свой путь шифровальщика. А можно зашифровать ценные данные через приложение *Seahorse*. Нажмите Alt+F2, введите **seahorse** и нажмите Enter.



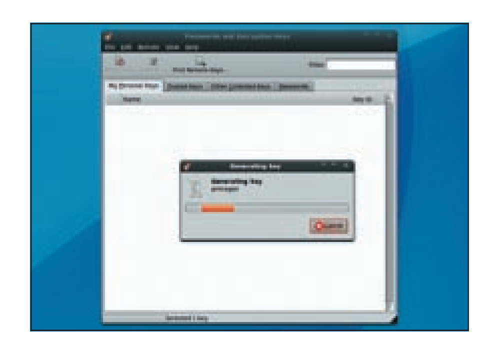
2 Определим ключ

Запустив *Seahorse*, выберите Файл > Создать [File > New] и укажите ключ PGP [PGP Key] в списке. PGP – это асимметричная система шифрования, позволяющая безопасно обмениваться файлами с друзьями. Нажмите Далее [Continue] и введите необходимую информацию – Имя [Name], электронный адрес [Email] и комментарий. Последний может быть напоминанием о том, для чего используется данный ключ.



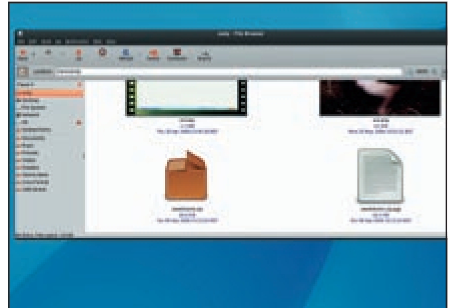
3 Проверим силу

Нажмите на небольшую стрелку Дополнительно [Advanced], чтобы просмотреть дополнительные опции. Мы не собираемся здесь ничего исправлять, но взглянуть стоит. Тип шифрования – это используемый способ кодирования содержимого. По умолчанию это алгоритм цифровой подписи (DSA – Digital Signature Algorithm) и его мощность – 2048 бит, что должно отогнать всех, кроме особо упертых хакеров, поскольку на его вскрытие требуется пара лет.



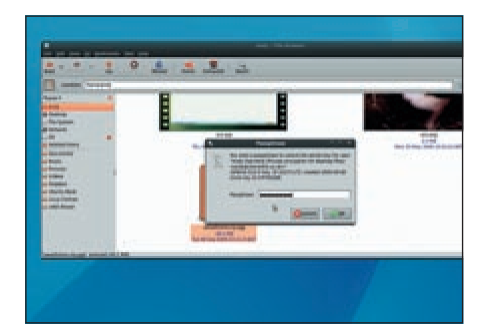
4 Шифруем файлы

Нажмите кнопку Создать [Create]. Придется немного подождать, пока создается ключ, а затем вы сможете добавить парольную фразу. Потом готовый ключ появится на вкладке Мои личные ключи [My Personal Keys]. Теперь вернитесь к каталогу, щелкните на нем правой кнопкой и выберите Зашифровать [Encrypt]. Отметьте в списке ключ и нажмите OK. Вам будет предоставлен выбор: шифровать каждый файл отдельно (потребуется вводить пароль для каждого файла) или в виде ZIP-архива.



5 Разберемся с файлами

Выберите шифрование с упаковкой, задайте имя и укажите расширение файла вроде .zip, .tar.gz или .jar и нажмите OK для начала процесса шифрования. В результате получатся два новых файла: первый – архивированная версия каталога, второй – тот же самый файл, но зашифрованный. Первый файл не защищен, поэтому его следует удалить.



6 Доступ к информации

Последний этап наступит когда-нибудь потом: вы захотите получить доступ к файлам еще раз. Просто поместите зашифрованный файл в то место, куда его следует распаковать, и дважды щелкните на нем. Будет запрошен пароль, и после его ввода появится новый каталог, готовый к употреблению. LXF

APT: Следим за

Вокруг нас – целый мир приложений; главное – держать их в свежем виде.

Одно из преимуществ Linux перед другими системами – то, что каждый дистрибутив имеет колоссальное количество приложений, специально упакованных и доступных за несколько щелчков мыши. Система управления пакетами в Ubuntu основана на *Advanced Package Tool* (*APT*) от Debian, который прекрасно выполняет работу по определению различных зависимостей, необходимых конкретному пакету, а затем устанавливает их в должном порядке. Например, вы можете пожелать установить Пакет А, но он основан на пакете В, а пакету В требуется новая версия пакета С. *APT* просканирует серверы Ubuntu и найдет обновленную версию пакета С перед установкой пакета В, а потом загрузит и установит пакет А, не требуя от вас знаний о библиотеках, DLL-файлах или какой-либо технической информации.

Возможно, вы уже видели подобные действия в системе автоматического обновления, которая изредка напоминает о необходимости обновить приложения из-за вышедших исправлений ошибок, повышения безопасности или добавления новых функций. Если вы еще не сталкивались с автоматическим обновлением, можете включить его и проверить наличие обновлений, выбрав Система > Администрирование > Менеджер обновлений [System > Administration > Update Manager]. На нашей тестовой системе менеджер обновлений сообщал, что машина находится в актуальном состоянии, но нажатие кнопки Проверить [Check] выявило, что имеется около 50 обновлений, от мелочей типа библиотеки Ogg Vorbis до вещей серьезных, вроде новой версии почтового клиента *Thunderbird*.

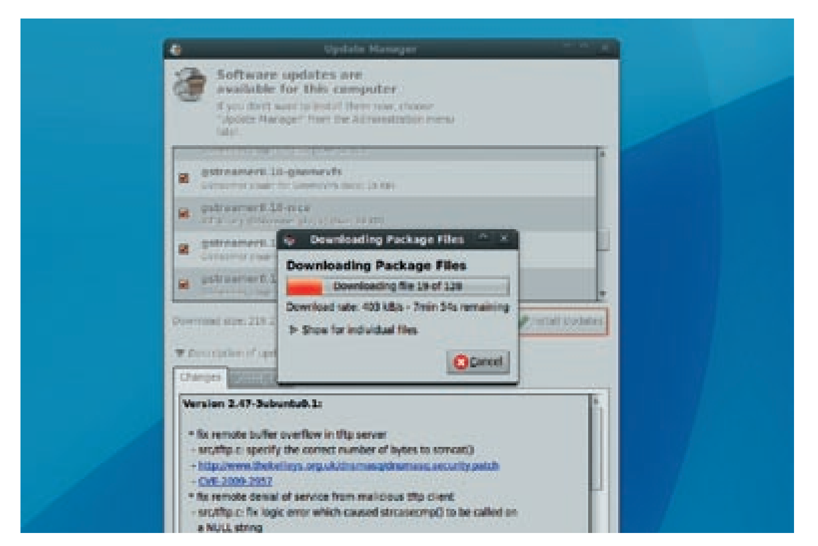
Чтобы начать обновление, просто нажмите кнопку Обновить [Update] и подождите, пока приложения загрузятся и установятся. Иногда (в ситуациях вроде установки нового ядра) по завершении обновления может потребоваться перезагрузка.

Менеджер обновлений становится действительно необходим после выхода новой версии дистрибутива: например, предстоит обновление до Karmic Koala (9.10). После официального выхода в менеджере обновлений появится новая кнопка «Сменить версию» [Upgrade] и уведомление о том, что доступна новая версия дистрибутива. Нажав ее, вы запустите полное обновление всей системы, что может занять значительное время и обязательно потребует перезагрузки.

Скорая помощь

Перед выполнением полного обновления системы позаботьтесь о наличии свежей резервной копии ваших данных.

› Даже если вы считаете, что ваша система актуальна, следует почаще убеждаться в этом.





› Добавить/Удалить — самый простой способ добавить новое приложение в настольный Ubuntu.

Добавляем приложения

Ubuntu имеет три различных метода добавления новых приложений. Первый – выполнять все в командной строке, что в принципе полезно; но графические инструменты много проще, и мы поработаем именно с ними. Простейший из них – пункт меню Добавить/Удалить [Add/Remove] в самом низу меню Приложения [Applications]. Он заполнен набором приложений, отобранных инженерами Ubuntu и размещенных в категориях, схожих с пунктами меню Приложения. От игр до графики, от утилит связи до калькуляторов – программы здесь охватывают все аспекты вашей компьютерной жизни.

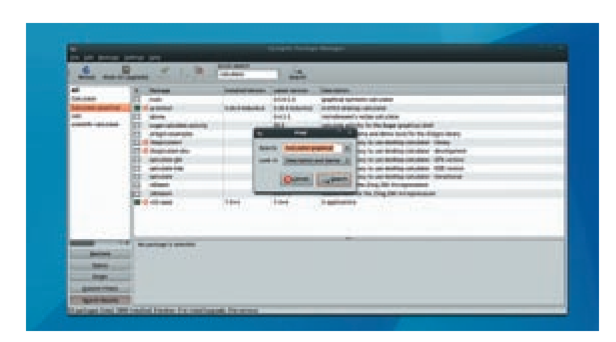
Вы можете воспользоваться панелью инструментов в верхней части окна, чтобы отфильтровать показываемые приложения или поискать определенный пакет или задачу. Щелчок на имени приложения в основном списке позволит изучить расширенную информацию и перейти на домашнюю страницу проекта. Столбец справа показывает популярность приложения – по числу установок – в виде рейтинга звездочками от 1 до 5.

Все элементы, рядом с которыми имеются галочки, уже установлены; вы можете снять галочку, чтобы поместить пакет в список на удаление, или щелкнуть на приложении без галочки, чтобы установить ее и поместить программу в список на установку. Это и вправду просто.

Завершив выбор приложений, жмите на кнопку Применить изменения [Apply Changes], чтобы запустить процесс установки и удаления. Как и в менеджере обновлений, приложения и их зависимости будут загружены с серверов Ubuntu и установлены. Они появятся в меню Приложения в соответствующем подразделе.

В разделе Добавить/Удалить [Add/Remove] имеются сотни приложений, но это лишь часть запасов Ubuntu. Следующий шаг, который раскроет вам тысячи приложений – *Synaptic*, гра-

обновлениями



› Диалог быстрого поиска позволяет подстраивать запрос, для получения более точных результатов.

фическая оболочка для вышеупомянутой системы APT. *Synaptic* можно найти в меню Система > Администрирование [System > Administration].

Он чуть посложнее, чем Добавить/Удалить, но работает по тому же принципу. Чтобы найти приложение, например, калькулятор, введите строку поиска в окне (в данном случае 'calculator') и просмотрите список программ, удовлетворяющих данному запросу. Затем можно доработать список, набрав уточняющую информацию. Например, ввод 'graphical scientific calculator' [графический инженерный калькулятор] удалит одни пункты и добавит другие.

Для таких случаев удобно нажать кнопку Поиск [Search], чтобы вызвать диалог, поскольку он допускает поиск по дополнительным критериям, таким как Название [Name]; Название и описание [Name and Description] и Хранитель [Maintainer]. Для нашего случая, 'graphical calculator' с опцией Имя и описание выдаст наиболее полезный список. Теперь мы можем поочередно выбирать каждый найденный элемент, чтобы получить точное описание приложения и, используя кнопку Скриншот [Get Screenshot], может быть, даже увидеть, как выглядит программа. Щелкнув на элементе правой кнопкой мыши и выбрав Свойства [Properties], вы получите дополнительную информацию. Откроется новое окно со вкладками и информацией о версии, зависимостях и т. д.

Как и в Добавить/Удалить, квадратик слева указывает, установлено ли приложение (если да, то он будет зеленым), но это не просто поле с двумя вариантами выбора: вы можете щелкнуть на нем, чтобы пометить приложение для Установки [Installation], Обновления [Upgrading], Удаления [Removal] или Полного удаления [Complete Removal]. Выбор пункта Удалить ликвидирует само приложение, но сохранит все его настройки, так что при установке в другой раз настройки пригодятся. Полное удаление убирает абсолютно все, давая вам чистый лист для переустановки программы. Определившись с выбором действия, нажмите на кнопку Применить [Apply] для запуска процесса.

Но каким бы умным ни был *Synaptic*, в конечном итоге все зависит от скорости одобрения новых пакетов инженерами Ubuntu: они дают вам доступ к свежим стабильным версиям приложений, а иногда этого не достаточно. Тут необходимо ломать барьеры и идти своей дорогой.

Обновление или установка приложения, не поддерживаемого официально, часто означает просто загрузку Deb-файла для конкретной версии Ubuntu, а затем вызов установщика двойным щелчком по нему. *APT* обычно берет зависимости на себя, и все будет готово. Однако загруженное приложение иногда требует зависимости, недоступные в Ubuntu. Поэтому многие разработчики предоставляют репозитории, содержащие обновления, необходимые для запуска их приложения, часто вместе с зависимостями.

Выходим за ворота

Чтобы воспользоваться ими, необходимо добавить репозиторий разработчиков в *Synaptic*. Этот несложный процесс начинается с запуска *Synaptic* и выбора Настройки > Репозитории [Settings > Repositories]. В появляющемся окне со вкладками необходимая секция озаглавлена Сторонние приложения [Third Party Software]. На стандартной установке она будет пуста, и репозитории необходимо добавить.

Нажмите кнопку Добавить [Add]. В открывшемся диалоге требуется вставить строку для APT, содержащую тип репозитория (двоичные пакеты или исходные тексты, обозначающиеся как deb и deb-src соответственно), URL сервера, обозначение дистрибутива и компонент.

Для примера, давайте настроим Ubuntu на установку разрабатываемой версии браузера Google *Chrome*. Строка *APT* будет иметь вид:

```
deb http://ppa.launchpad.net/chromium-
daily/ppa/ubuntujaunty main
```

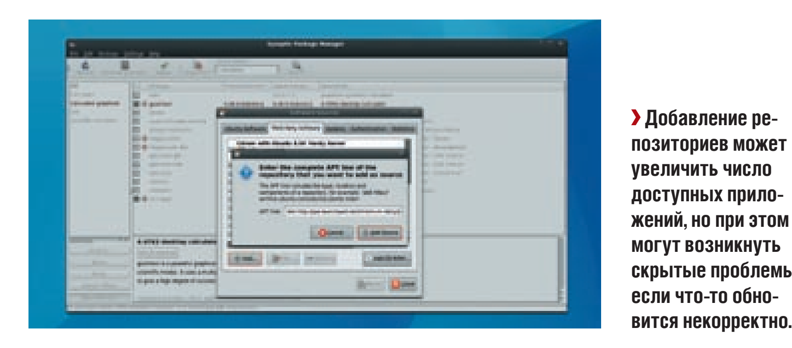
Если вы работаете в старой версии Ubuntu, измените обозначение с intrepid или hardy (или karmic, если вы уже установили последнюю версию), а затем нажмите кнопку Добавить источник [Add Source]. Наконец, нажмите Закрыть [Close].

Перед выполнением любых изменений необходимо нажать кнопку Перезагрузить список [Reload], чтобы заставить *Synaptic* скачать свежую информацию о добавленном репозитории. По завершении, найдите приложение и пометьте его для установки обычным способом.

Аналогично можно удалить репозитории или воспользоваться кнопкой Редактировать [Edit], чтобы изменить информацию. Хорошо бы иногда просматривать репозитории, чтобы выяснить, все ли добавленное вы еще используете; и если нет, удалить ненужные записи. Это особенно верно в случае, если репозитория больше не существует. LXF

Скорая помощь

Если вы используете приложения из репозиториев разработчиков, вам, возможно, следует обновляться почаще, в связи с быстрым темпом развития приложений.



› Добавление репозиториев может увеличить число доступных приложений, но при этом могут возникнуть скрытые проблемы, если что-то обновится некорректно.

» **Через месяц** Обновляем SUSE и тестируем *Chrome*, новый браузер от Google.

GIMP: В стиле

На дождливом уроке этого месяца художник и разработчик **Майкл Дж. Хэммел** преобразится во Фрэнка Миллера. И однажды дождь пойдет по-настоящему...




До


После

Наш эксперт

Майкл Дж. Хэммел участвует в разработке *GIMP* и является автором трех книг по данной тематике, включая самую новую – «Эффекты в *GIMP*: руководство для художников».

В детстве я не особо зачитывался комиксами издательств Marvel или DC. Я часами упивался картинками из «Звездного пути» и фотографиями Луны, сидел на крылечке за чтением Кларка и Азимова, рвался к звездам вместе с Карлом Саганом и прятался под одеялом, чтобы послушать радиопередачи д-ра Даменто. Я был странным ребенком, даже по меркам читателя комиксов. Но с тех пор я вырос и, если верить моей жене, выгляжу гораздо представительнее среднего технаря. Я меньше читаю, больше смотрю кино и вполне прилично могу сыграть басовую партию в *Guitar Hero*. И по-прежнему не интересуюсь комиксами.

Однако даже взрослым, но все еще чокнутым типам вроде меня практически невозможно полностью игнорировать влияние, которое комиксы и их мрачные собратья, графические новеллы, имеют в обществе. В постоянных попытках все более изуверским способом пустить кровь своим персонажам кинопродюсеры обращаются к комиксам, переводя сюжеты этого мирка на язык большого экрана. Если пренебречь жуткими приключениями Спауна и Кролика Роджера, Голливуд, похоже, нашел свою комикс-нишу и счастливо влился в мир супергероев.

Героям — слава!

Так что придется отдать должное жанру. Недавние шедевры «Железный человек» и «Темный рыцарь» добавили «чернухи» в воплощения Супермена и Чудо-женщины, известные с 1970-х годов. Город Грехов [Sin City] и Хранители перенесли это новое видение в еще более мрачные пределы. Город Грехов, с его высокой контрастностью и пятнами цвета, произвел удовлетворительное художественное впечатление на меня как на фаната *GIMP*, что в итоге послужило основой сегодняшнего урока...

На сей раз нам понадобится несколько стоковых фото, чтобы показать, как наш оборванный герой под покровом сырой и неприглядной ночи обходит дозором криминальные районы развращенного города. Все будет черно-белым, с единственным пятном цвета. Самое приятное в работе с черно-белым изображением – то, что оно скорей всего не выйдет из диапазона цветов местного струйного принтера. Единственный момент, когда это может быть не так – работа с цветовым пятном.

Ресурсы этого урока

» **Стоковые фото:**

Мужик с пушкой [Man with Gun] www.bigstockphoto.com/photo/view/2123798
Мужик с пушкой (галстук) [Men with Gun (Red Tie)] www.bigstockphoto.com/photo/view/3343608
Пистолет [Handgun] www.bigstockphoto.com/photo/view/2944267
Строение [Building] www.morguefile.com/archive/?display=55516

» **Месяц назад** Мы расслаблялись, монтируя мир любви и взаимопонимания.

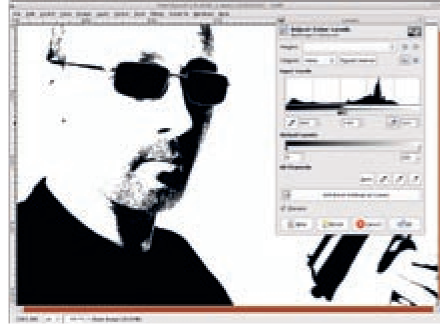
Города Грехов

Шаг за шагом: Создаем картинку в стиле «гризайль»



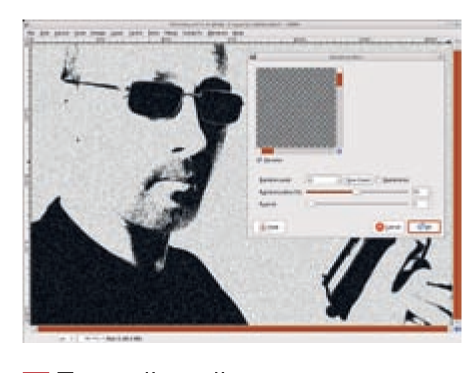
1 Базовое изображение

Базовое изображение предоставляет нам главного героя, «крепкого орешка» с пистолетом, на однотонном светлом фоне с разрешением 1600×1067 пикселей. Обесцветим изображение (Цвет > Обесцветить...), выбрав опцию Светлота, что сохранит высокий контраст между лицом и фоном.



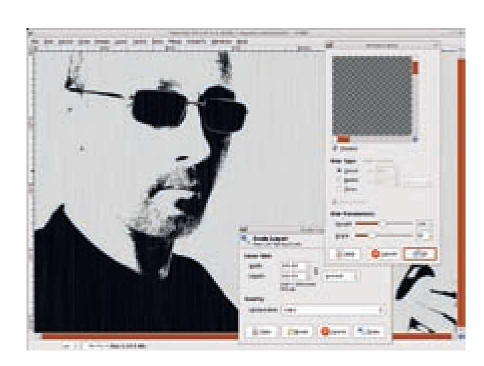
2 Увеличим контраст

Откроем диалог Уровни (Цвет > Уровни), затем щелкнем на пипетке точки белого (с правой стороны под гистограммой Уровней) и щелкнем в правом верхнем углу фона изображения. Это сделает фон полностью белым, слайдер точки белого подвинется влево от центра гистограммы. Передвинем слайдер точки черного вправо почти до самой точки белого. В итоге получится практически черно-белое изображение.



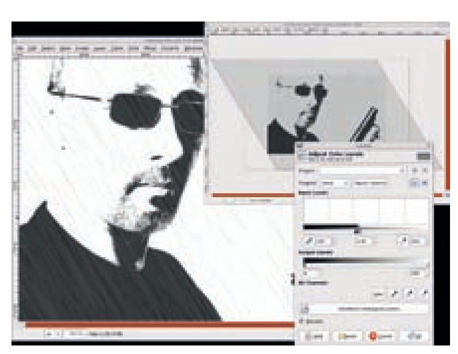
3 Первый слой дождя: шум

Создадим прозрачный слой (Слой > Создать слой) с именем Дождь1. Откроем фильтр Бросок (Фильтры > Шум > Бросок). Значения по умолчанию тут вполне подойдут: Случайное зерно = 10, Случайность %= 50, Повторов = 1. Нажмем OK, чтобы визуализировать шум в слое Дождь1. Шум будет окрашенным, и мы его обесцветим (я выбрал здесь опцию Светлота, но подойдет любой из способов обесцвечивания).



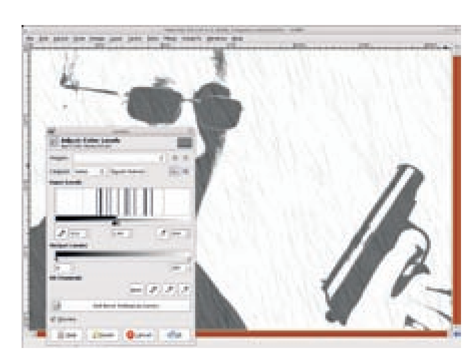
4 Первый слой дождя: капли

Откроем фильтр Размывание движением (Фильтры > Размывание > Размывание движением), выберем «Линейное», длину 120 и угол 90 градусов. Щелкнем OK, чтобы применить размывание к слою Дождь1. Увеличим размер слоя на 40 %, то есть изменим единицы измерения в диалоге Смена размера слоя (Слой > Размер слоя) на проценты и выставим 140 в поле ширины. При нажатии кнопки Изменить высота изменится автоматически, чтобы соблюсти пропорции.



5 Первый слой дождя: перекос

Уменьшим масштаб, чтобы увидеть границу слоя Дождь1, затем выберем на панели инструментов инструмент Искривление. Щелкнем на изображении и протащим его, пока края слоя не коснутся правого верхнего и левого нижнего углов изображения, затем применим изменения щелчком по кнопке Искривить. Откроем диалог Уровни и настроим точку белого примерно на значение 107, а точку черного – примерно на 116. Восстановим размеры слоя (Слой > Слой к размеру изображения).



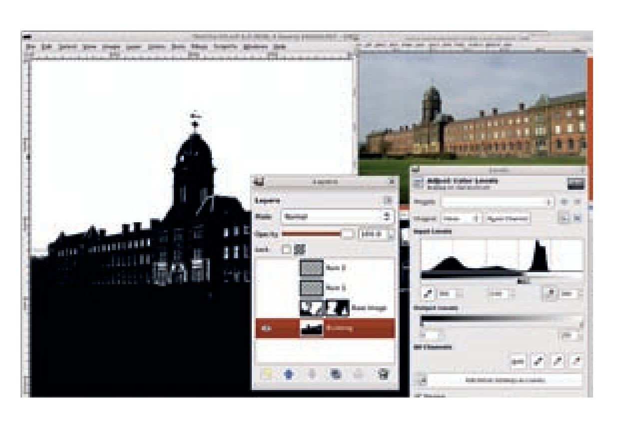
6 Второй слой дождя

Аналогичным образом добавим второй слой с дождем, но применим не такое сильное Искривление, чтобы дождь падал более вертикально. Настройка Уровней также может быть несколько другой, на ваш вкус. Позже мы подправим режимы обоих слоев и настройки непрозрачности, поскольку контраст изменится из-за добавления фоновых элементов. Назовите второй слой Дождь2. »



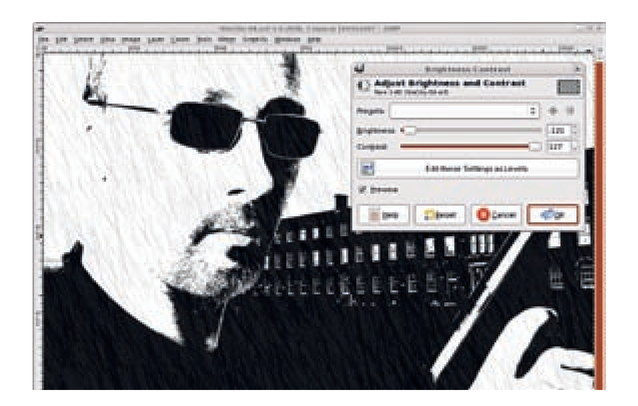
7 Маскируем базовое изображение

Щелкнем по слою с базовым изображением, чтобы сделать его активным. Добавим ему альфа-канал и белую маску, затем снова откроем исходное изображение. Выберем инструмент Выделение связанной области, выставим значение порога на 70 и сделаем щелчок при нажатой клавише Shift, чтобы выбрать области, окрашенные голубым. Вставляем выделение в маску слоя базового изображения. Через диалог уровней выставим точку черного на 254.



8 Здание на заднем плане

Изменим размер фонового здания так, чтобы оно соответствовало размеру слоя базового изображения, затем вставим его в базовое изображение и переместим новый слой в самый низ стопки слоев. Отключим видимость всех других слоев, чтобы облегчить следующие шаги, затем обесцветим и настроим контраст изображения точно так же, как мы это делали на Шаге 2.



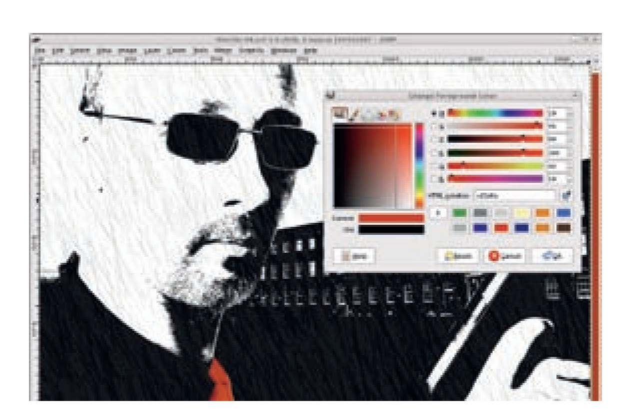
9 Обработка слоев с дождем

В диалоге слоев уменьшим непрозрачность каждого слоя с дождем до 30 %, затем сольем эти два слоя (щелчком правой клавишей мыши на слое Дождь1 и выбором «Объединить с предыдущим»). С помощью диалога Яркость-контраст настроим контраст объединенных слоев (Цвет > Яркость-контраст), выставив значение яркости на -121 и контрастности – на 127.



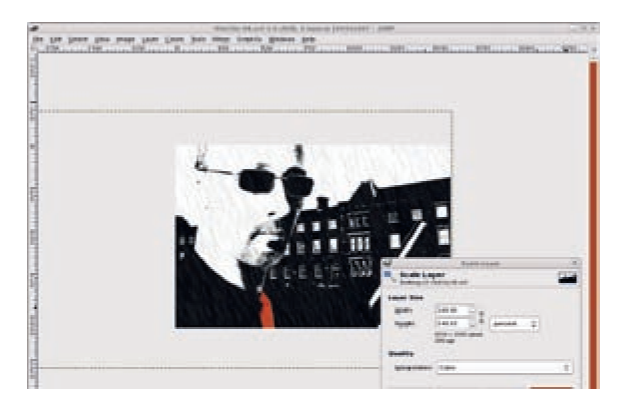
10 Добавим цветовое пятно

Откроем изображение гангстера с красным галстуком. Выберем инструмент Выделение связанной области и выставим значение порога на 95. Щелкнем в любом месте красного галстука. Если внутри выделения останутся невыделенные области, увеличьте выделение на один пиксель, чтобы захватить их. Иллюстрация показывает выделение с задействованной Быстрой маской зеленого цвета. Скопируем выделение.



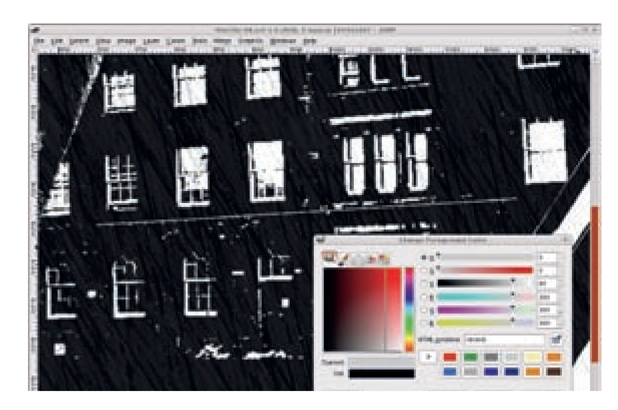
11 Повяжем галстук

Вставим выделение в базовое изображение, зеркально повернем его по горизонтали, затем обесцветим слой и выберем новый красный цвет. Отметим галочкой опцию «Запереть альфа-канал» в диалоге слоев. Выберем инструмент Заливка, настроим режим «Объединение зерна» и область применения «Заполнить все выделенное». Щелкнем на слое Красный галстук, чтобы перекрасить изображение.



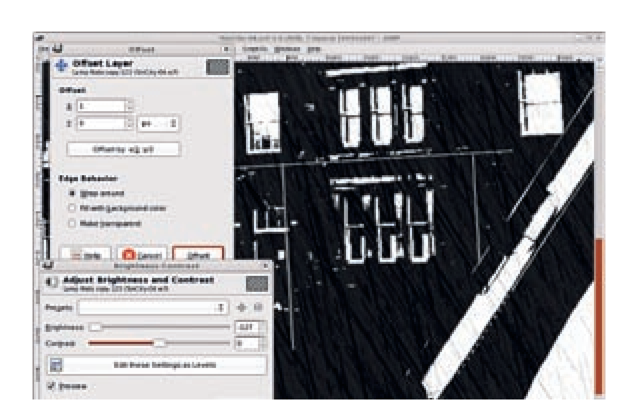
12 Изменяем здание

Откроем диалог изменения размера слоя (Слой > Размер слоя). Щелкнем по иконке цепочки, чтобы «разорвать» и изменить соотношение пропорций слоя. Выставим ширину слоя на 165 % и высоту – на 140 %. С помощью инструмента Перемещение перетащим слой влево, пока парадный вход здания не окажется слева от пистолета. Снова изменим размер слоя, чтобы он совпал с размером изображения. (Слой > Слой к размеру изображения).



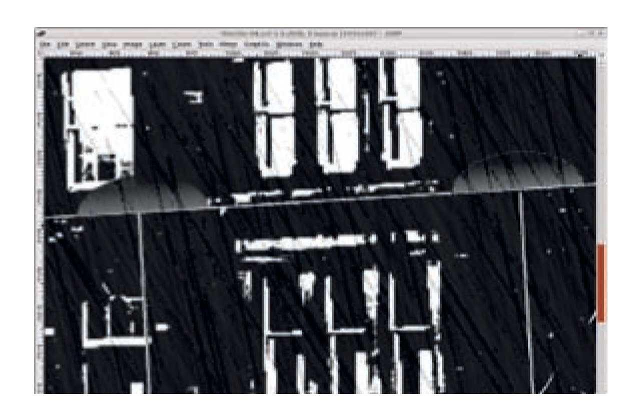
13 Направляющая для фонарей

Добавим прозрачный слой прямо над слоем со зданием. Установим цвет переднего слоя в светло-серый (HTML-код – cbcbcb). Проведем линию слева направо примерно на уровне верха второго этажа здания. Позднее мы удалим этот слой.



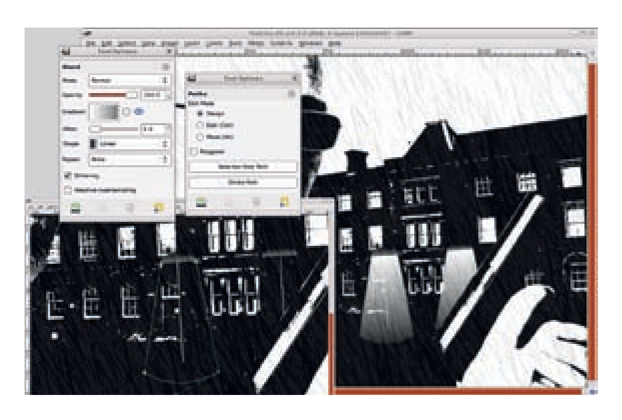
14 Рисуем фонарные столбы

Добавим прозрачный слой над слоем с направляющей и кистью малого радиуса нарисуем линию слева от главного входа от направляющей вниз к фундаменту здания. Повторим процесс справа от входа. Скопируем этот слой и сместим его на один пиксель по оси X. Уменьшим яркость до -127.



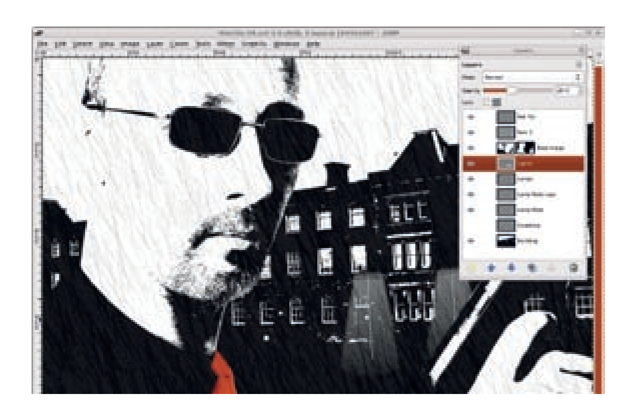
15 Фонари

Создадим овальное выделение в новом слое, затем отрежем нижнюю половину выделения. Расположим выделение над левым столбом. Выберем инструмент Градиент и протащим линейный градиент от черного к серому, от верха выделения прямо вниз. Повторим действие для другого фонаря, затем уберем выделение (Выделение > Снять).



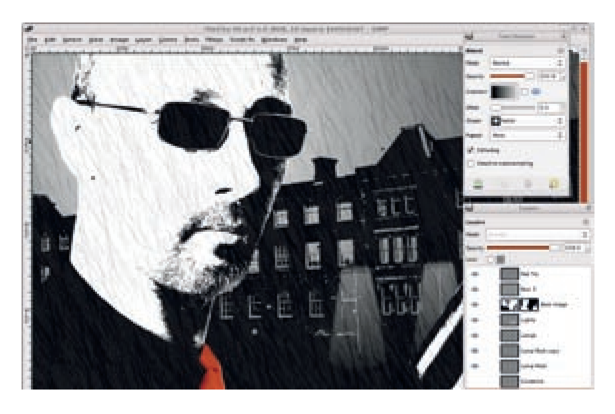
16 Свет

Создадим в диалоге Слои прозрачный слой под слоем фонарей. Инструментом «Кривые» очертим вставку под левым фонарем, в выделении применим градиент от белого к прозрачному сверху вниз и повторим эти действия для другого фонаря. Уберем выделение.



17 Настроим свет

Откроем фильтр Гауссово размывание (Фильтры > Размывание > Гауссово размывание) и применим размывание в 10 пикселей к обеим осям (Y и X) слоя фонарей. Установим непрозрачность слоя между 35 и 40 %. На данном этапе можно удалить слой с направляющей линией или же просто отключить его видимость в диалоге слоев.



18 Финальный эффект

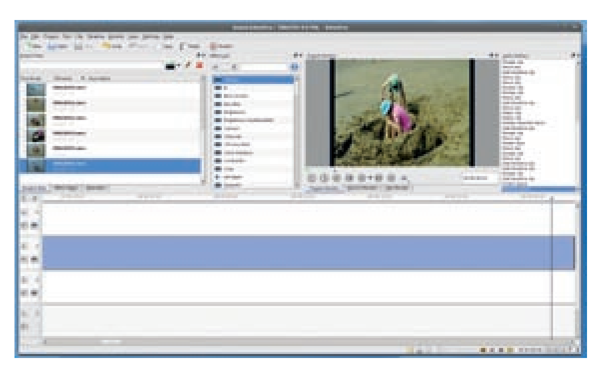
Чтобы затемнить фон, добавим прозрачный слой сразу над слоем со зданием, зальем его черным, добавим маску слоя и выберем инструмент Градиент. Настроим радиальную форму от черного к серому, затем сделаем щелчок посреди лица мужчины и протащим к его указательному пальцу на пистолете, чтобы создать радиальную маску. LXF

» **Через месяц** Приукрасим рабочий стол иконками в стиле «а-ля OS X».

Kdenlive: Чистим

Часть 2: Энди Ченнел обрисует для вас редактирование видео в *Kdenlive* – отбор, монтаж и добавление эффектов – на примере создания короткого фильма.





Используйте инструменты Масштаб и Высота дорожки для улучшения предпросмотра содержимого вашего видеоряда.



Наш эксперт

Энди Ченнел всю дорогу снимает автобиографию на видео – на старости лет он заново прокрутит свои звездные часы.

На прошлом уроке мы насобирали видеоклипов и выполнили черновой монтаж видеопроекта с помощью *Kdenlive*. Теперь облагородим полученный продукт инструментами *Kdenlive*, добавив эффекты и переходы, а в завершение снабдим его титрами.

При создании коротких видеороликов вроде нашего редактирование не особенно затруднительно, но чтобы сделать все правильно, потребуется время. Прелесть пакетов типа *Kdenlive* в том, что можно все менять, не боясь угробить исходный видеоряд. Благодаря эффективности файлового формата *Kdenlive* также возможно создать много различных «нарезок» из одного и того же материала с помощью Файл > Сохранить как [File > Save As]. Однако помните, что ролики, уже импортированные в программу, нельзя перетаскивать в другое место в *Nautilus*, *Dolphin* или *Konqueror*: *Kdenlive* запоминает только пути к файлам, но не их содержание. Редактирование – процесс творческий, и каждое сделанное исправление несет определенный смысл. Наша цель сейчас – обеспечить ощущение места и события. Дикторский текст нас здесь не особенно волнует.

> «Можно все менять, не боясь угробить исходный видеоряд.»

Мы уже сделали примерный монтаж того, что нам нужно отредактировать, и теперь, на первой стадии процесса, подрежем и наслоим отдельные клипы ради лучшего расчета времени всего куска. Как и прежде, простой способ это сделать состоит в наведении указателя мыши на одну из границ клипа, а затем щелчке и перетаскивании, чтобы сократить или увеличить кусок.

Два быстрых изменения сделают этот процесс более интуитивным. Во-первых, воспользуйтесь маленьким значком у основания экрана для отображения видеоминиатюр. Тогда будут видны первый и последний кадры клипа. Затем приближайте с помощью ползунка Zoom [Масштаб] и установки Высоты дорожки, которые находятся над органами управления треком. Две иконки увеличивают либо уменьшают высоту, обеспечивая удобный предпросмотр содержания. Это особенно пригодится позже, когда мы будем редактировать звук.

Осторожно, разрыв!

При обрезке клипов в вашей временной шкале образуются пробелы, потому что клипы становятся короче, а последующий участок остается на том же месте. А значит, нужно широко применять инструмент Spacer [Раздвижка], который находится в маленькой панели у основания окна, рядом с инструментом Cutter [Лезвие].

Щелкните по нему, и указатель мыши станет стрелкой вправо – это указывает, что он воздействует на объект, который вы выберете, и все, что находится на временной шкале справа от этого объекта. Например, если вы сократили время выполнения клипа 1, на временной шкале появится разрыв между ним и следующим клипом. Инструментом Spacer выберите клип 2 и перетащите влево, чтобы передвинуть остальные ваши клипы как один объект.

Удовлетворившись продолжительностью и размещением клипов, принимайтесь за другие способы обработки, в том числе оформление переходов с клипа на клип, редактирование звука и добавление привлекательных эффектов. Первая из этих вещей, также известная как монтажный переход, основана на неболь-

Месяц назад Мы освоили *Kdenlive* и подготовили видеоматериал.

LINUX FORMAT на DVD
» Kdenlive 0.7.5

продукт

шом наложении соседних клипов: по мере исчезновения одного возникает второй. Начните с перетаскивания второго клипа нашей исходной пары – тех самых клипа 1 и клипа 2 – слегка влево. Помните, что мы используем два трека на шкале времени, так что клип 1 лежит на треке 0, клип 2 – на треке 1, клип 3 – снова на треке 0 и т.д. Длина наложения определяет продолжительность перехода, но ее можно редактировать, поэтому чрезмерно о точности не заботьтесь.

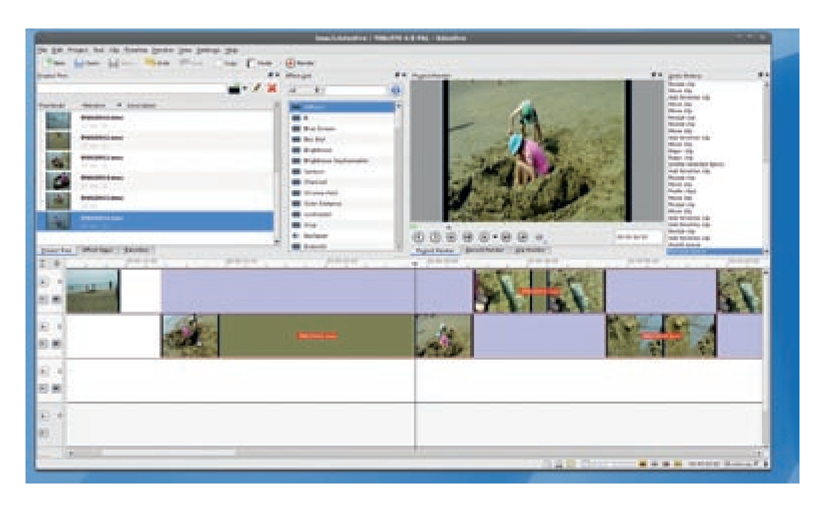
Чтобы создать сам переход, наведите мышь на нижний правый угол верхнего клипа, щелкните правой кнопкой и выберите пункт Добавить переход > Наплыв [Add Transition > Dissolve]. Появится желтое окно, перекрывающее участок наложения треков, с названием перехода в нем. Щелкните по окну и просмотрите настройки перехода, перейдя на вкладку Стойка эффектов [Effects Stack]. Вы увидите выпадающее меню с типом перехода и возможностью указать любой доступный трек. При выборе Auto [Автоматически] система будет работать со смежными дорожками.

Опций для настройки перехода не так уж много, но одна из них довольно важна – это переключатель, обозначенный Reverse [Обратно]. Мы заметили, что иногда все работает нормально в первый раз, при использовании наплыва от дорожки 0 к дорожке 1, а потом как бы «ломается». Это происходит потому, что при переходе с трека 0 на 1 (или 1 на 2 и т.д.) должна быть выбрана опция Reverse, но при переходе в другую сторону ее надо убирать.

Убейте меня нежно

Длительность перехода регулируется точно так же, как и продолжительность клипа, но учитывайте иерархичность отображения: верхний трек имеет приоритет. Например, если вы потянете правый край наслоения клипов 1 и 2 влево, вы в итоге перейдете к клипу 2, но затем будет резкий скачок к клипу 1, а затем другой – к клипу 2. Более того, если вы перетащите этот правый край еще дальше, наплыв нормально завершится, а потом зритель увидит резкую (и неприятную) смену кадров. Хитрость тут в том, чтобы подобрать время перехода и затем, при необходимости, снова подрезать клипы, чтобы все было гладко.

В пакете доступно несколько разных переходов, и вы, быть может, соблазнитесь использовать в вашем видео их все. Однако они могут отвлекать от смысла отснятого, поэтому старайтесь не переборщить.



› Наплыв мягко переводит от одной сцены к другой.

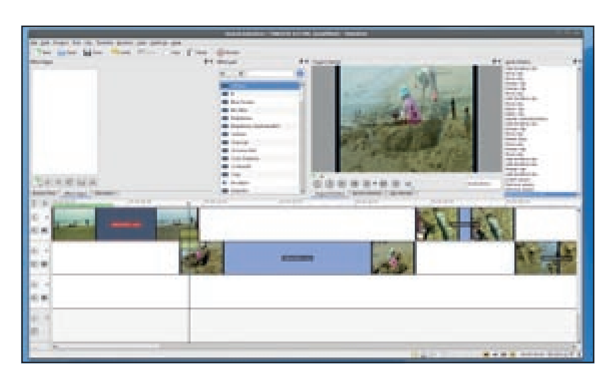
Хотя в нашем видео мы позаботились о переходах, можно заметить, что звук в клипах все еще прыгает, оставляя нехорошее впечатление. К счастью, затухание аудио между двумя клипами можно добавить быстро, всего лишь несколькими щелчками мыши. Можно организовать усиление (fade-in), при котором звук постепенно становится громче, поводив указателем мыши над верхним углом на начале клипа, пока он не замигает зеленым, а затем щелкнув и перетащив вправо, чтобы задать скорость процесса. Ослабление (fade-out) добавляется так же, но вам следует перейти к концу клипа и тащить влево. Перекрестное затухание (cross-fade) достигается за счет синхронизации ослабления звука одного клипа с усилением второго (как на экранном снимке внизу).

Плохое качество видео зрители могут терпеть долго, но этого нельзя сказать о плохом качестве аудио. В конце концов, глаза всегда можно закрыть; а вот не слышать уже труднее. Поэтому, шлифуя ваш звук, необходимо потратить по крайней мере столько же времени, сколько вы потратили бы на визуальные эффекты.

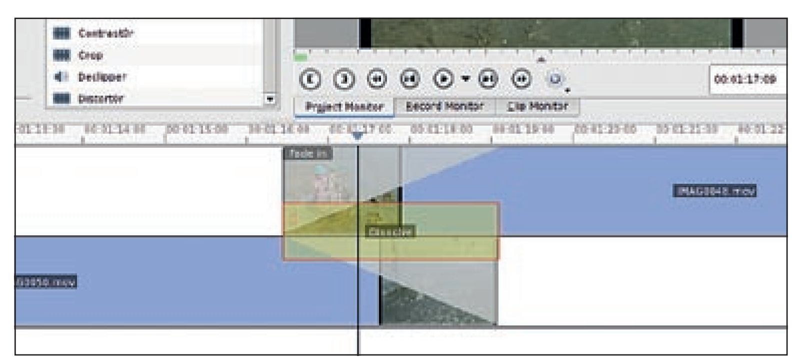
В материале для нашего урока существует значительная разница между звуковым сопровождением кадров, снятых стоя перпендикулярно морю и перегруженных шумом ветра, и тех, что были сделаны против ветра или в укрытии. Разница очень заметна, поэтому необходимо действовать по схеме «ослабь и замени». »



За время между публикациями части 1 и части 2 этого учебника *Kdenlive* обновился до версии 0.7.5, которую мы используем на нашем уроке. Это важное обновление: оно включает (как эффект) ключевые кадры, а также исправляет немало ошибок.



› Инструмент Spacer делает длинные череды клипов более управляемыми. Он перемещает все, что находится справа от курсора.



› Наплывы сгладят визуальные переходы, но неплохо позаботиться и о затухании звука между клипами, чтобы не терзать слух ваших зрителей.

Когда вы импортируете клип в *Kdenlive*, по умолчанию аудио соединено с видеофайлом. Это удобно для монтажа и редактирования, но не дает доступа отдельно ко звуку, поэтому наша первая задача заключается в выявлении клипа, содержащего дребезжание, правому щелчку мыши по нему и выбору пункта Разделить аудио [Split Audio]. На первой доступной звуковой дорожке создастся новый клип с требуемым звуком. Теперь можно будет менять его, не трогая визуальных эффектов.

› **При работе с аудио иногда удобно отключить видеоряд, чтобы ничто не отвлекало.**

Ваш аудиогид

Далее следует точно определить, где именно понадобится правка – это можно сделать, прислушавшись к звуку и расставляя метки там, где вы хотите начать и закончить редактирование. Чтобы добавить метку, щелкните по полоске в верхней части временной шкалы и выберите пункт Добавить направляющую [Add Guide]. Вы можете снабдить метку комментарием, например, «Здесь добавить быстрое затухание»; затем нажмите кнопку OK. В этом месте во всех дорожках появится сплошная линия.

Добавьте нужное количество меток и начинайте редактирование. Мы могли бы пойти тем же путем, что и при обработке видео, но затухание звука удобнее выполнять по огибающей его уровня. Для этого щелкните по аудиоклипу правой кнопкой и выберите Добавить аудиоэффект > Уровень звука (с опорного кадра) [Add Audio Effect > Volume (Keyframeable)]. Примерно около двух третей высоты клипа появится белая полоса – она представляет уровень звука. Затухание обеспечивается путем двойного щелчка на ней с последующим перетаскиванием получившейся управляющей точки вниз или вверх, для уменьшения или увеличения громкости. Кроме того, можно сократить или удлинить время затухания перетаскиванием управляющей точки влево или вправо. В этом процессе нам показалось удобным перейти на показ аудиоминиатюр, через настройки в основании временной шкалы.

Чтобы убрать шум ветра из нашего материала, мы назначили опорный кадр для ввода затухания, как только начался шум; но тогда возник участок, где звук был слишком тихим. Поэтому мы перетащили клип с хорошим звуком в конец временной шкалы, щелкнули правой кнопкой мыши и выбрали Разделить аудио для создания двух файлов. Нас интересовал только звук, так что мы щелкнули правой кнопкой мыши на видео еще раз и выбрали Разгруппировать клипы [Ungroup Clips] (можно также нажать Ctrl+Shift+G), чтобы разорвать связь между двумя частями. Тогда стало возможным выбрать видео и нажать Удалить [Delete], чтобы избавиться от него.

Наконец, щелкните и перетащите аудиораздел к месту разрыва и перетащите его вниз, на звуковую дорожку два. Здесь следует подумать о продолжительности и содержании, потому что выбранный звук должен вписываться в сцену и быть достаточно длинным для покрытия разрыва. Укажите позицию, а затем добавьте Fade In и Fade Out (или другую опорную огибающую, если вы в ней уверены) и установите их таким образом, чтобы уровень звука сочетался с содержанием.

Хотя основная сила *Kdenlive* состоит в редактировании видео, доступность параметров для настройки звука означает, что вам не придется отправляться за этим в *Audacity*. Инструменты монтажа и огибающей, упомянутые здесь, не исчерпывают перечень эффектов: их можно добавить в ваш звук через пункт контекстного меню Добавить аудиоэффект. Процедура добавления эффектов такая же, как для видео, к чему мы сейчас и переходим.

> «Затухание звука удобнее выполнять по огибающей.»

Наложение эффектов

Как и переходы, эффекты можно использовать для украшения вашего материала. Тут довольно легко погорячиться и нашпиговать эффектов там, где они в общем-то ни к чему. Итак, поймав себя на наведении мыши на эффект Oldfilm, задайте два вопроса: какой смысл эффект внесет в мое видео, и скроет или подчеркнет он то, что я хотел сказать своим фильмом? Если ответы будут позитивными, тогда – вперед.

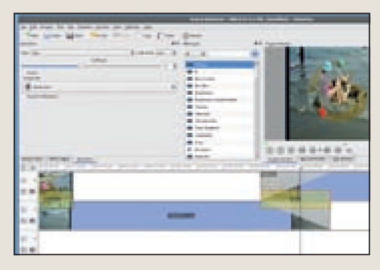
Kdenlive работает по принципу Стойки эффектов: это список процессов, применяемых к клипу на входе в определенном порядке. Чтобы добавить эффект, выберите клип, а затем перейдите к Линия времени > Добавить видеоэффект > … [Timeline > Add Video Effect > …]. Это можно сделать и через контекстное меню. В обоих случаях эффект будет применяться к клипу и заноситься в стойку. От очередности эффектов зависит конечный результат, поэтому предусмотрены инструменты для изменения этого порядка. Например, на экранном снимке справа мы применили к клипу три эффекта: Greyscale [Оттенки серого], Oldfilm [Старомодный], Scratchlines [Царапины]. Эта комбинация создает впечатление, будто фильм снят на старую пленку. Однако фильтр

Тотальное затемнение

Другой основной переход, который вы, скорее всего, не обойдете стороной – это Wipe [Затемнение]. Он используется в кино многие десятилетия, и вы можете вспомнить его применение для обозначения смены места действия в «Звездных Войнах» – дань Джорджа Лукаса Акире Куросава. Добавление Затемнения в *Kdenlive* – процесс, аналогичный наплыву, и точно так же время перехода определяет скорость Затемнения. Попробуйте выбрать Wipe как тип перехода, а затем используйте раскрывающееся меню Файл изображения [Image File], чтобы выбрать один из видов Затемнения из списка.

Это вовсе не статические изображения: скажем, настройки Radial-bars покажут серию концентрических окружностей, появляющихся в клипе, которые постепенно растут, пока не начнется второй клип. Ползунок Резкость [Softness] в верхней части раздела определяет контраст между краями двух клипов и поможет вам делать довольно приятные мягкие переходы.

Как мы уже отметили, с переходами легко переборщить, так что применяйте их экономно.



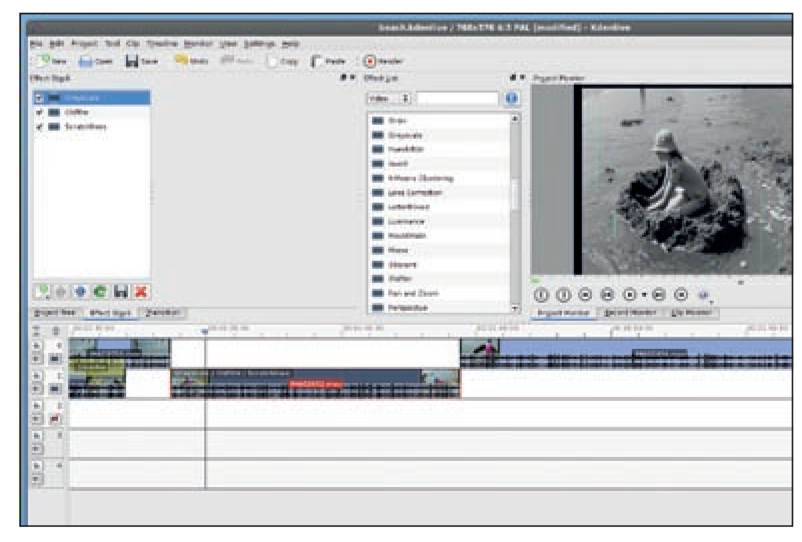
› **Спиральное затемнение — приятный эффект, если им не злоупотреблять.**

Scratchlines предназначен для имитации царапин на цветной пленке, и штрихи получатся тоже цветные. Чтобы обойти эту проблему, изменим порядок эффектов, с тем, чтобы Scratchlines применялся первым, а затем преобразуем все в монохром эффектом Greyscale, создав более правдоподобный вид.

В основании окна Стойки эффектов помещен набор иконок; с их помощью можно добавлять новые эффекты, изменять их порядок или сбрасывать его в исходный и сохранять эффект для дальнейшего применения. Последний из этих вариантов особенно полезен, если вы настроили эффект и хотите использовать его для следующих клипов. В этом случае, нажмите кнопку Сохранить [Save], укажите имя для своего эффекта и затем применяйте его так же, как любой другой готовый эффект.

Программа включает массу эффектов, в том числе простые фильтры настройки яркости, контраста или цвета клипа, но также простирается до более экстремальных вариантов, пригодных для изоляции цвета, создания яркой и контрастной картинки в мультфильме или пикселизации части экрана.

Разумное использование нескольких эффектов может реально улучшить видео, но в равной степени можно чересчур увлечься и испортить его.



› Зеленые штрихи неуместны в черно-белом фильме, и правильный порядок эффектов требует применения фильтра Greyscale последним.

Дразнящие титры

Завершим редактирование, добавив в начале фильма название; сделаем это в два этапа. Сперва создадим кадр с титрами, добавим его на временную шкалу и применим подходящий стиль. Начните с перехода в Проект > Добавить клип титров [Project > Add Title Clip]. В верхней части этого окна находятся три важных инструмента, которые – слева направо – позволяют перемещать элементы, создавать текстовые поля и рисовать прямоугольники. Справа от них предусмотрено несколько параметров, вид которых зависит от контекста, а под ними рабочая зона. Там можно много чего понаделать, но мы-то просто создадим заголовок в нижней трети кадра, который будет маячить в первом из наших клипов.

Выберите инструмент Прямоугольник [Rectangle] и изобразите прямоугольник вдоль нижнего края рабочей области. После этого нажмите кнопку Заполнить цветом [Fill Color] и определите цвет окна. Теперь ползунком рядом с этой кнопкой определите прозрачность прямоугольника. Мы задали его черным, с прозрачностью 66 %.

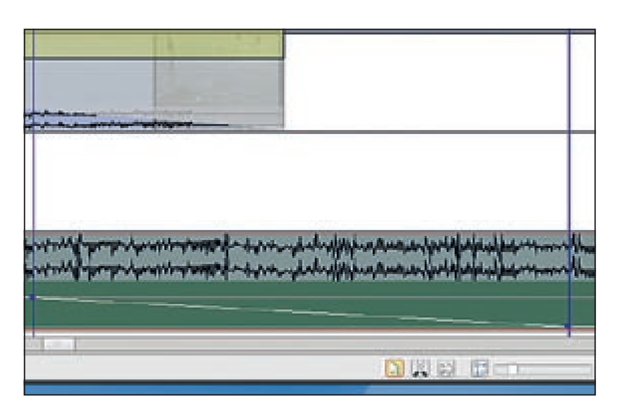
Выберите инструмент Текст [Text] и щелкните где-нибудь внутри прямоугольника. Теперь введите заголовок и при помощи инструментов в верхней части окна определите его шрифт, размер и цвет. Мы выбрали белый текст на прозрачном черном фоне и шрифт без засечек, чтобы это легко читалось с экрана телевизора. Обратите внимание, что вы можете изменять порядок элементов – очередность их слоев – используя параметр Z-INDEX в правом верхнем углу. Чем выше это число, тем ближе объект к вершине стойки слоев.

Щелчком по значку Картинка [Picture] можно добавить в титры изображение: он открывает ваш стандартный обозреватель файлов. Для масштабирования и правки картинки используйте инструменты данного окна. Закончив, нажмите кнопку OK; создадутся титры, которые будут добавлены в ваше дерево проекта.

> «Редактирование – вещь долгая и требует свежего взгляда.»

Теперь можете перетащить готовое название на временную шкалу для интеграции в ваше видео. Поскольку мы используем двухдорожечное редактирование и первый клип у нас на треке 0, перенесем титры на трек 1 в начало первого клипа. После перетаскивания титры появятся со своим собственным эффектом Composite. Однако сейчас он настроен на неправильный трек, поэтому щелкните и перетащите его, чтобы оно было на обоих треках 0 и 1. Отметьте флажок Composite и измените Операцию альфа-канала [Alpha Channel Operation] в выпадающем меню на XOR.

Титры являются регулируемыми, как и все прочее, то есть можно, например, установить эффект затухания вдоль всего клипа, но это довольно муторная операция, и на нее у нас здесь не хватает места.



› Используйте опорные кадры со звуком для создания огибающей, контролирующей уровень громкости в ваших аудиодорожках.

Все дело в отделке

Мы собрали здесь видеоряд с переходами и сложным звуком, а также несколькими эффектами и титрами, вводящими зрителя в курс дела. Редактирование – вещь долгая, если вы хотите сделать все верно, и оно также требует свежего взгляда. Итак, удовлетворившись видеорядом, дайте ему отлежаться и вернитесь к нему через несколько дней – вы поймете, вправду ли он великолепен. Не горюйте, если он вам разонравится и придется снова браться за работу: при первой попытке редактирования вы многому научились.

На следующем уроке мы выполним финальное сохранение в файл в различных форматах и создание образа DVD, пригодного для воспроизведения на домашнем плейере. LXF

» **Через месяц** Прокрутим наш фильм и сделаем DVD для домашнего просмотра.

Blender: Содвинем

Часть 3: Наша серия уроков подходит к своему логическому концу. По этому поводу можно и бокал шампанского налить, считает ее автор **Андрей Прахов**.

LINUX FORMAT на DVD
» Модель этого урока



Наш эксперт

Андрей Прахов
Участник нескольких игровых проектов, представитель СМИ, редактор электронного журнала «Linux Games Technologies».

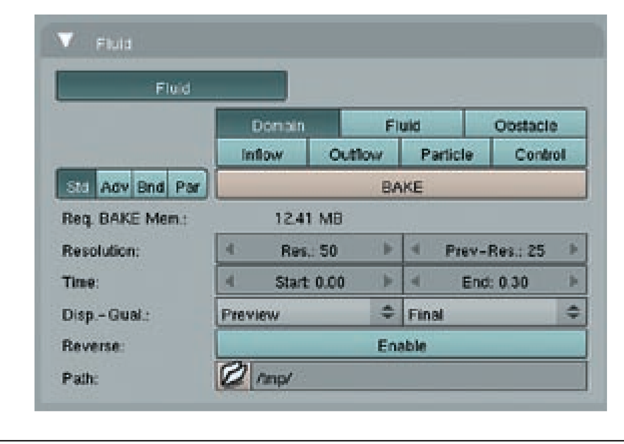
На прошлых занятиях мы с вами сервировали небольшой столик для романтического ужина. Расставлены фужеры, горят свечи, льется мягкий свет, но чего-то не хватает! Правильно: весь вечер сидеть и любоваться на пустые бокалы – это занятие не для нас. Так давайте наполним их вином!

Немного теории

Blender, как и любой другой мощный инструмент трехмерного моделирования, умеет работать с физикой жидкостей. Данная возможность появилась относительно недавно, если говорить не о времени, а о номерах версий программы. Следствием этого является слабая документированность. Пара-тройка примеров – вот и все, на что вы можете рассчитывать на официальном сайте. Но и этого достаточно, чтобы разобраться в имеющихся настройках. Другой вопрос в том, что в процессе имитации может участвовать не всякий объект, и некоторая доработка «напильником», а иногда и «танцы с бубнами» будут вам обеспечены. Но хватит о грустном: в конце концов, предназначение этой статьи и есть в «подстилании соломки» ищущим знания.

Так как эта часть урока отводится теоретическому вступлению в физику жидкостей, то забудьте на время об имеющемся проекте «Романтический ужин», а просто вчитывайтесь и иногда экспериментируйте в новой сцене.

Основополагающим краеугольным камнем физики жидкости в *Blender* является понятие «домен» [Domain]. Этим термином обозначается некоторая область сцены, в пределах которой будут производится необходимые расчеты. Все, что не попадает в домен, в процессе имитации участвовать не будет. Это правило распространяется и на те объекты, которые оказались в домене некоторой своей частью. Например, если наш фужер поставить на условной границе домена и «наполнить» жидкостью, то она остановится ровно по границе. В качестве домена может выступать любой Mesh-объект, но так как физика жидкостей работает только с ограничивающим параллелепипедом [bounding box], то логично использовать примитив Cube.



› **Рис. 1. Настройки режима Domain.**

Еще одним обязательным компонентом является объект-источник [Fluid]. Здесь никаких ограничений на формы объекта нет: единственно, он должен быть из группы Mesh.

Рассмотрим, как взаимодействуют оба компонента. При запуске просчета объект-домен принимает форму объекта-источника и постепенно перетекает на первоначальное место своего расположения. Сложно представить? Тогда немного попрактикуемся.

Создайте новый проект и переключитесь в окно просмотра Camera (0 на цифровой клавиатуре). Уже имеющийся в центре куб послужит нам в качестве домена. Отмасштабируйте его до желаемого размера. Теперь добавьте в сцену сферу и разместите ее так, чтобы она находилась внутри куба. Для удобства манипулирования переключитесь в режим Wireframe клавишей Z.

Отлично, теперь приступаем к настройке физики. Выделите Cube и нажмите уже известную вам кнопку Physics Buttons (**LXF124**). На сей раз нас интересует панель Fluid с единственной пока одноименной кнопочкой. Включите ее.

Появятся восемь дополнительных кнопок, позволяющих указать режим имитации для выделенного объекта. Естественно, нас интересует кнопка Domain (рис. 1).

Так как это основные настройки физики жидкости, рассмотрим их подробнее.

Первое, что бросается в глаза – это крупная кнопка с названием Bake [Выпечка]. Не думайте, что достаточно настроить физику жидкостей, нажать заветные Alt+A – и *Blender* резво начнет анимацию действия в сцене. В отличие от уже рассмотренных ранее частиц и физики мягких тел, просчет жидкостей необходимо выполнить заранее. Именно эта кнопка активирует так называемую «выпечку мешей». Перед ней необходимо подготовить место для кэша, указав путь в поле Path. По умолчанию *Blender* использует каталог **/tmp**.

На окончательное качество картинки влияют следующие настройки:

» **Resolution** Здесь поле Res позволяет выбрать необходимое разрешение для рендера, а поле Prev-Res – для окна программы. Конечно, желательно использовать несколько большее значение Res, чем предлагаемое по умолчанию. Внимание: даже незначительное увеличение выходного разрешения на какой-нибудь десяток единиц приведет к серьезной нагрузке на систему; к тому же излишне высокое разрешение негативно сказывается на качестве имитации!

» **Disp.-Qual** Выбор режима качества для окна программы и рендера, соответственно.

Пока оставьте настройки домена по умолчанию.

» **Месяц назад** Физика мягких тел и частиц = бархат и пламя свечей.

бокалы!



› Рис. 2. Результат работы тестового примера.

Теперь займемся объектом-источником. Выделите сферу, откройте панель Fluid и нажмите кнопку Fluid. Опций здесь минимум, но они чрезвычайно важны для настройки течения. Группа кнопок Volume Init позволяет определить, как жидкость будет обтекать имеющийся объект в домене:

» **Volume** По умолчанию активна именно она. Жидкость обтекает вокруг объекта.

» **Shell** Если объект имеет полую форму (к примеру, чашка), включите эту кнопку.

» **Both** Режим, объединяющий первые два.

По умолчанию, жидкость всегда стремится вниз по координате Z. Если нужно иное направление, то измените переменные в группе Initial Velocity.

В нашем случае, оставьте все как есть. Выделите еще раз объект-домен и нажмите кнопку Bake (рис. 2).

Дорабатываем сцену

Как видите, работа с жидкостями не вызывает особых проблем. Но возникает вопрос, какой смысл во всей этой системе, если изначально домен может иметь только форму куба? Ответ прост: помимо рассмотренных двух компонентов, в процессе имитации могут участвовать и другие объекты с иными настройками физики. С ними мы познакомимся при доработке нашей главной сцены. Конечно, она не предполагает быть анимационной, но полученные знания, надеюсь, вам впоследствии пригодятся.

В создаваемой сцене мы наполним шампанским имеющиеся два бокала. В качестве источника на этот раз используем сферу, немного сплюснутую по координате Z. Такие сферы необходимо будет расположить непосредственно в чашечках обоих бокалов. Но перед настройкой физики необходимо доработать сами модели.

Как уже говорилось ранее, механизм физики жидкостей очень чувствителен к качеству структуры объектов в домене. Особенно это касается полых моделей. В первой статье мы рассмотрели один из способов создания бокала с помощью кривых NURBS. То, что получилось в итоге, выглядит красиво, но для участия в процессе имитации не годится.

Выбираем тип жидкости

Жидкость жидкости рознь: к примеру, масло характеризуется повышенной вязкостью. Не зная физических свойств нужной жидкости, очень сложно добиться ее качественной имитации. К счастью, разработчики программы подготовили несколько заготовок, которые можно найти, если нажать кнопочку AD (Advanced) на панели Fluid. По умолчанию *Blender* предлагает использовать воду (water).

Давайте разберемся, что произошло. Выделите бокал, затем нажмите кнопку Z для перехода в режим Wireframe и обратите внимание на исказившиеся формы модели. В структуре появились пустоты, напрочь исчезло дно. Если такую модель использовать далее, то жидкость просто выльется «на пол». В то же время, объект в режиме редактирования и заливки (Solid) выглядит абсолютно нормальным. Интересно, что эта «особенность» программы не влияет на результат отрисовки.

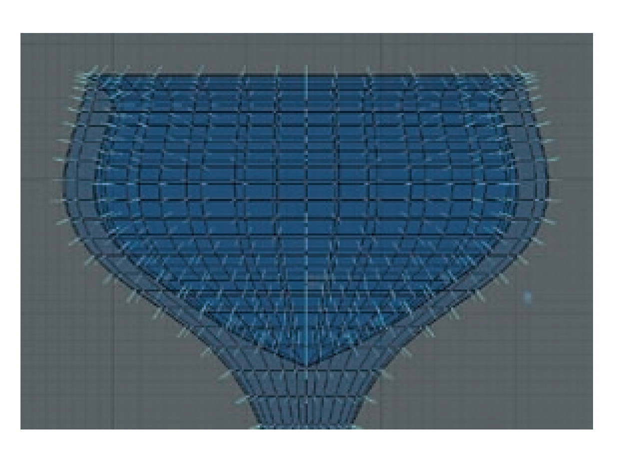
Разгадка неоднозначного поведения *Blender* заключается в неправильной конвертации из NURBS в Mesh. Именно на этом этапе происходит разрыв структуры. Есть только одно средство для борьбы с этой ошибкой: использовать меньшее разрешение NURBS.

Выделите модель (формата NURBS!) и нажмите F9 для открытия панелей редактирования. В опциях закладки Curve Tools имеются два окошка: Resol U и V. Именно они отвечают за количество ребер в структуре объекта. Необходимо, пошагово уменьшая значения полей, пробовать конвертировать объект в Mesh. Дефект структуры может проявится не сразу, поэтому после конвертации попробуйте перейти в режим редактирования и обратно. В моем случае, оптимальным выбором значений стало U = 3, V = 2.

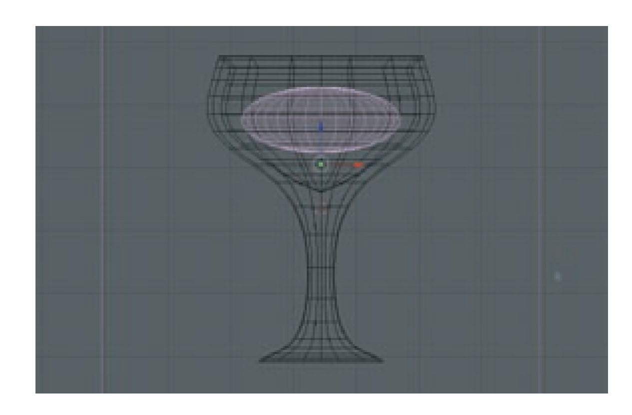
Естественно, конвертацию необходимо производить только в режиме просмотра Wireframe, чтобы сразу заметить разрывы и «откатить» назад. Все остальные действия по доработке бокала остаются неизменными (**LXF123**).

Основное правило при работе с полыми объектами гласит: все нормали вершин должны смотреть наружу. Это является обязательным условием для правильной работы физики жидкости.

По умолчанию, любой примитив, создаваемый программой, всегда имеет нормали, повернутые наружу. Но так как мы дубли-



› Рис. 3. «Нормальные» нормали должны быть повернуты наружу!



› Рис. 4. Расположим на сцене несколько вспомогательных объектов.

ровали чашечку бокала для имитации толщины стенок, то нормали копии смотрят вовнутрь. В этом несложно убедиться, если заставить *Blender* показать на экране нормали объекта (рис. 3).

Выделите бокал и войдите в режим правки. Нажмите F9 для открытия панелей редактирования. Во вкладке Mesh Tools More нажмите кнопку Draw Vnormals, и программа прорисует нормали всех вершин объекта. Для изменения направления нормалей воспользуйтесь пунктом меню Mesh > Normals > Recalculate Outside любого окна 3D View при выделенной структуры объекта. Обязательно посмотрите, что получилось. Иногда программа «пропускает» некоторые вершины. В этом случае придется изменять направление нормалей вручную.

Вот и все, доработка бокала закончена.

Настраиваем физику

Все остальные действия лучше проводить в режиме Wireframe, для полного контроля над происходящим.

Сейчас нам нужно добавить домен, разместить объект-источник, настроить физику. Выделите бокал и с помощью функции Snap окна 3D View (Object > Snap > Cursor → Selection), переместите курсор в середину объекта. Добавьте куб и сферу, как показано на рис. 4.

Выполним настройку физики за несколько шагов:

1 Сфера Так как этот объект является источником, включите кнопку Fluid. Теперь его можно убрать в любой свободный слой, для исключения из результата обработки сцены (клавиша m).

2 Бокал Здесь нам понадобиться функция Obstacle. Именно она позволяет объектам взаимодействовать с жидкостью в качестве препятствий. Модель бокала является полой, поэтому включите добавочную кнопку Shell.

3 Куб Естественно, он является доменом. Активируйте соответствующую кнопку.

Настройте по своему усмотрению путь для кэша и смело жмите кнопку Bake (рис. 5).

Работаем с материалом

То, что у нас получилось, очень мало смахивает на вкусное шампанское. Не будем проводить аналогии с другим неаппетитным содержимым, а просто доработаем материал.

Создайте для объекта-домена новый материал с базовым цветом R = 1.000, G = 1.000, B = 0. Затем откройте панель Shaders. В качестве диффузного шейдера оставьте имеющийся по умолчанию Lambert. Увеличьте для него параметр Ref до 1.000. Отражающим шейдером будет Wardiso, дающий резкий, яркий отблеск. Установите следующие значения: Spec = 1.600, RMS = 0.077. В принципе, для этого материала нужно включить кнопку TraShadow ради качественной прорисовки тени, но это сильно скажется на времени обработки. Активируйте по желанию.

Теперь займемся преломлением и отражением. Откройте вкладку Mirror Transp и включите кнопку Ray Transp. В качестве коэффициента преломления для обычной воды при комнатной температуре подойдет значение 1.33 (IOR). Прозрачность, параметр Fresnel, установите в 2.5. Желательно несколько увеличить глубину просчета Depth. Я установил значение 5. Также нужно немного настроить зеркальность. Включите кнопку Ray Mirror и установите следующие значения: RayMir = 0.5, Fresnel = 2.0.

Если на данном этапе вы попробуете обработать сцену, то получите нечто, напоминающее подкрашенную водичку. Правильно, настоящее шампанское не может быть без пузырьков... ну разве что выдохшееся. По-хорошему, воздушные пузырьки нужно моделировать с помощью частиц. Этот способ оптимален для анимационной сцены. Но ведь у нас она статичная!

Пойдем несколько иным путем, гораздо менее затратным по времени, но по качеству не уступающим моделированию частиц, а именно: воспользуемся технологией Normal Map и процедурной текстурой.

Добавьте новую текстуру в первый слот материала. В качестве ее типа укажите процедуру Voronoi. Бóльшую часть параметров можно оставить по умолчанию, за исключением масштаба. Нам необходимо минимально возможное разрешение, поэтому сбросьте данные в окошке Size на ноль.

Выделите объект-домен, нажмите TAB, а затем U для установки текстурных координат (контекстное меню UV Calculation > Unwrap). Это необходимо для правильного проецирования текстуры. Откройте панель Map Input и включите кнопку UV. В последнем случае мы заставили Blender использовать текстурные координаты объекта.



› Рис. 5. Жидкость, но уж больно неаппетитная: скорее «отработка», чем шампанское.

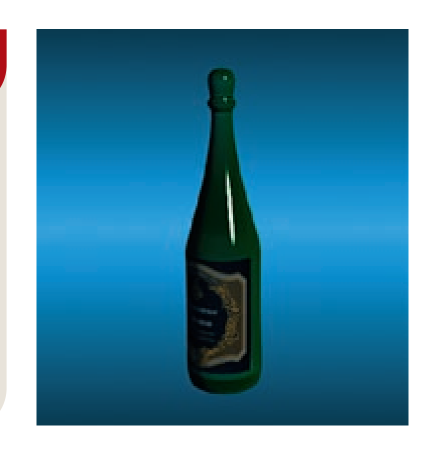
› Рис. 6. Процедурная текстура успешно имитирует пузырьки воздуха.

Используйте фоновое изображение

Если у вас наметанный взгляд и вы в детстве с легкостью рисовали натюрморты с оригинала, то следующий текст можете смело пропустить. Всем остальным при моделировании советую использовать возможность вывода картинки в качестве фона в окнах программы. В этом случае становится проще «накидать» каркас объекта по уже имеющемуся рисунку.

Для управления фоновым изображением необходимо выбрать пункт меню View > Background Image любого окна 3D View. В открывшейся плавающей панели можно загрузить собственно файл и настроить прозрачность и масштаб. Заметьте, что фоновое изображение доступно только для стандартных режимов просмотра (Front, Side, Top, Camera). Удобно, что при масштабировании сцены в окне изменяется и размер картинки. При окончательной обработке сцены фон не учитывается.

Эта панель способна на большее, нежели просто управление статичной картинкой. К примеру, вы можете использовать в качестве фона видеофайл.



› Рис. 7. Бутылка шампанского.

Настройки Normal Map доступны в панели Map To. Дважды нажмите кнопку Nor. Она должна стать желтого цвета, что сигнализирует о включении инверсионного режима наложения карты нормалей. Глубину выдавливания желательно поставить поменьше, к примеру, Nor = 1 (рис. 6).

Модель бутылки

Моделирование такого объекта не должно вызвать у вас никаких затруднений. В предыдущих статьях уже описывались два способа, поэтому выберите наиболее понравившийся. Здесь же постараемся сделать результат более реалистичным.

Допустим, что наша бутылка сделана из темно-зеленого стекла с фирменной этикеткой какой-нибудь винодельческой фирмы. Предположим, что вы уже создали модель и готовы приступить к ее «раскраске».

Добавьте новый материал со следующим основным цветом: R = 0.151, G = 0.343, B = 0.096. В панели Shaders выберите Fresnel и Wardiso. Самостоятельно добейтесь яркой поверхности материала с жестким отблеском. В настройках прозрачности Ray Transp установите IOR равным 2.00. Значение параметра Fresnel нужно подобрать так, чтобы стекло стало полупрозрачным: к примеру, попробуйте 1.5.

Теперь займемся наложением этикетки на бутылку. На прилагаемом к журналу диске уже имеется необходимая картинка в формате PNG. Если вы захотите использовать свою, то учтите два правила:

» Разрешение картинки должно быть достаточно большим.

» Изображение должно быть с альфа-каналом в местах фона. Соответственно, формат файла должен поддерживать альфа-канал (PNG, Targa, TIFF...).

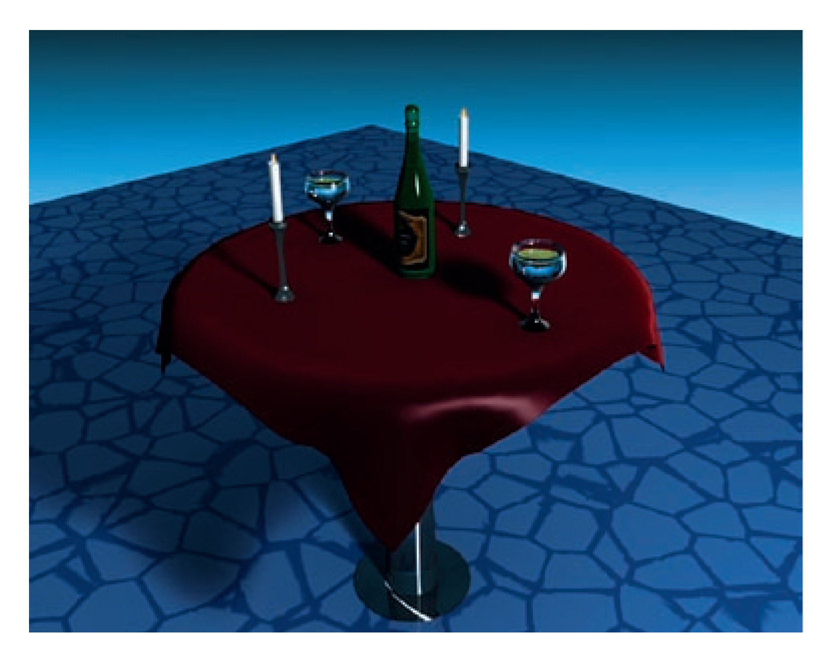
Перенесите модель в пустой слой для удобства работы с ней и откройте в соседнем окне UV/Image Editor. Загрузите текстуру в свободный слот материала (тип Image). Проследите, чтобы в панели настройки картинки Map Image была включена кнопка Use Alpha. Кроме того, необходимо изменить настройки рендера окон 3D-View для отображения в них текстур. В меню Game главного окна программы выберите пункт Texture Face Materials.

Нам необходимо включить развертку UV-координат для модели. Выделите объект в 3D-окне и нажмите кнопку Tab. Отметьте всю структуру объекта (клавиша A) и нажмите U для вызова контекстного меню UV Calculation. Выберите пункт Reset. Если теперь поменять тип прорисовки окна на Textured (меню Draw Type), то *Blender* станет отображать загруженные текстуры.

Итак, мы создали для модели текстурные координаты. Теперь займемся собственно проецированием картинки. Войдите в режим редактирования объектов и в окне UV/Image Editor выберите из списка уже загруженный файл. Выделите на объекте те полигоны, которые необходимо «залить» этикеткой, и выберите Unwrap в меню UV Calculation. Как видите, в окне UV/Image Editor появилось это же выделение. Теперь, используя стандартные возможности манипуляции (те же самые, что и для трехмерных объектов), разверните и отмасштабируйте заливку на картинке. Возникает логичный вопрос: а что же делать с остальной неиспользуемой частью бутылки? Все просто: выделите ее, примените Unwrap и разместите эту область на свободной от этикетки картинке, т.е. на фоне (рис. 7).

Готовимся к обработке

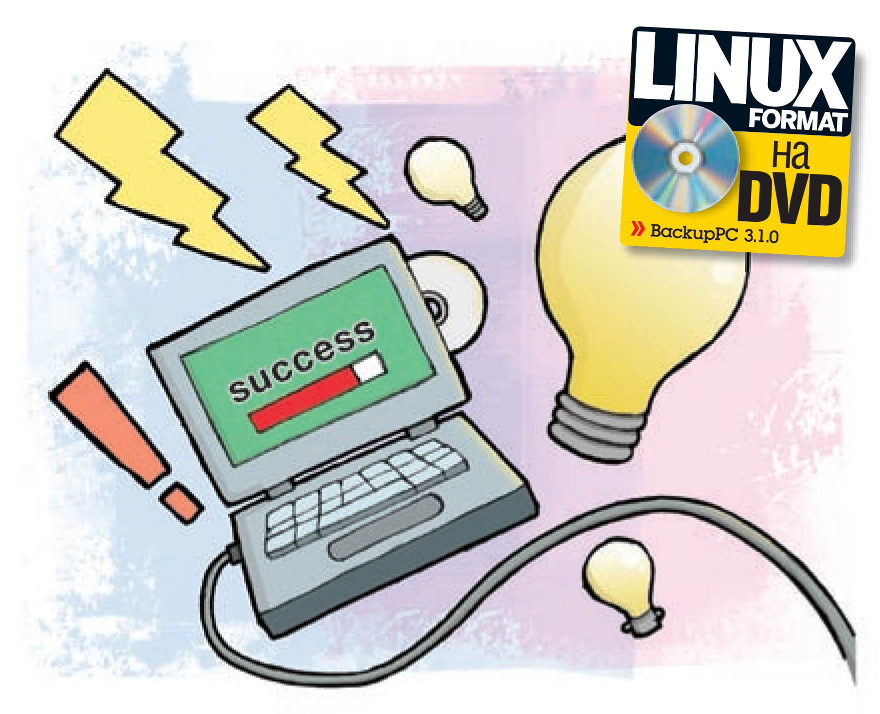
Для обработки такой сложной сцены с несколькими физическими имитациями нужно учитывать определенные правила. Во-первых, удалите все кэши, созданные программой. В некоторых случаях это обеспечит отсутствие ошибок. Установите счетчик кадров анимации на ноль. После окончательного просчета можно будет выбрать наиболее интересный кадр. Обработку физики следует начинать с просчета жидкостей. Как обычно, выделите домен и нажмите кнопку Bake. Только после этого можно нажать Alt+A для создания кэша частиц и физики мягких тел. Установите желаемый кадр, перейдите на панель рендера, выберите подходящий тип из заготовок – к примеру, PAL. Жмите кнопку Render и наслаждайтесь результатом (рис. 8)! LXF



› Рис. 8. Несколько штрихов — и картина закончена.

BackupPC: Как

Часть 12: Делать резервные копии нудно, но **Нейл Ботвик** поможет вам взвалить этот тяжкий груз на плечи сервера всего несколькими строками кода.





Наш эксперт

Нейл Ботвик
У Нейла Ботвика по компьютеру в каждой комнате. Но по соображениям безопасности он ни за что не скажет вам, где центральный сервер.

К сожалению, потеря данных – горькая правда жизни, и стопроцентно надежных носителей на свете нет. Потому и бытует поговорка, что все люди делятся на два типа: те, что не делают резервные копии, и те, кто их *уже* делают.

Тем не менее, вряд ли кому доставляет удовольствие торчать у компьютера, скармливая ему DVD-болванки. Машины созданы, чтобы упрощать жизнь, и как нам освежать копии важных данных, если нет ни времени, ни желания этим заниматься? Мы здесь говорим о сетях, и дело еще осложняется тем, что нужно хранить резервные копии данных с нескольких компьютеров, причем на некоторых из них может даже не быть графического интерфейса.

Разве не здорово было бы завести программу, которая следила бы за тем, когда и для каких компьютеров созданы резервные копии, и обновляла бы их в нужный момент, и все это без особых запросов? А еще лучше было бы уметь восстанавливать данные, частично или полностью, с любого компьютера сети, даже если целевому компьютеру стало настолько худо, что он грузится только с Live CD! К счастью, это не мечта – *BackupPC* (http://backuppc.sourceforge.net) может делать все это и даже больше. Для создания резервных копий с Linux и Mac она применяет *rsync*, а для резервных копий Windows – SMB (протокол разделения файлов Windows). Она помнит и про отсутствующие компьютеры, ожидая, пока те повторно включатся в сеть, и даже возобновит процесс резервного копирования после прерывания. Мы возьмем текущую версию *BackupPC*, 3.1.0. Версия 3.2.0 сейчас в процессе бета-тестирования, и она еще не готова для новых пользователей.

Начальная настройка

BackupPC состоит из двух основных частей: сервера, который занимается созданием резервных копий других компьютеров сети (и себя самого), и web-интерфейса для проверки резервных копий и восстановления данных из них. Последний работает через стандартный web-сервер *Apache*. Лучше всего добавить *BackupPC* через менеджер пакетов дистрибутива, если в репозитории есть соответствующий пакет. Установить его нетрудно, но так как это программа на Perl, она требует множество модулей, и список зависимостей почти такой же длинный, как и список отговорок, которые можно придумать, чтобы пока не делать резервную копию.

Следующий этап – настройка сервера. Здесь нужно принять во внимание три файла: **/etc/BackupPC/hosts** содержит список компьютеров, для которых создаются резервные копии, **/etc/BackupPC/config.pl** содержит глобальные настройки резервного копирования, а в файле **/etc/BackupPC/pc/имя_хоста.pl** эти настройки переписываются (или добавляются новые) для каждого компьютера. Продумайте также, где хранить резервные копии. Места потребуется много, но не так много, как кажется. Видите ли, когда *BackupPC* сохраняет файл, то проверяет, есть ли он уже в хранилище резервных копий, и если да – создает на него жесткую ссылку вместо сохранения новой копии. Итак, один у вас компьютер или сто, память, занятая операционной системой, будет все той же. Когда вы примете во внимание данные, ситуация может измениться, но значительная экономия места все равно получится.

> «BackupPC проверяет, сохранен ли уже файл, экономя место.»

Резервные копии лучше всего хранить на отдельном диске, особенно если вы планируете создавать резервную копию для того же компьютера, на котором запущен *BackupPC*. В файле **config.pl** можно изменить каталог, используемый для хранения данных, но это может вызвать непредвиденные побочные эффекты, поэтому оставьте его как **/var/lib/BackupPC** и смонтируйте туда свою файловую систему, выделенную для резервной копии.

В файле **hosts** перечисляются компьютеры, резервные копии данных с которых нужно создать, в следующем формате:

```
hostname dhcp-flag username
```

Здесь hostname, как вы уже догадались, это имя хоста, а флаг DHCP нужно установить в **0**, если у компьютера статический

» **Месяц назад** Мы писали и читали в Web, вооружившись WebDAV.

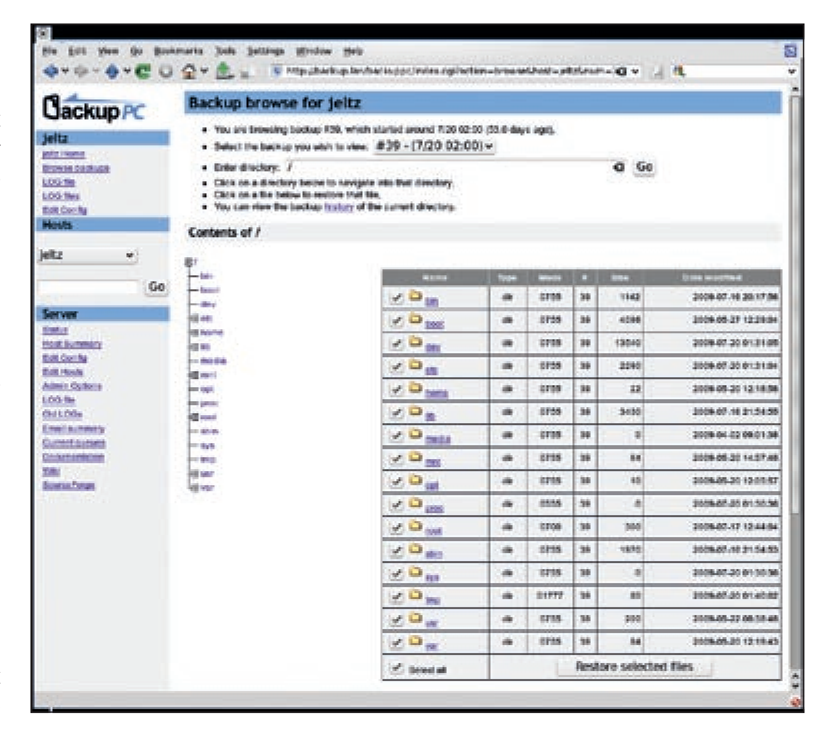
решить вопрос

адрес или его можно найти через Windows NMB, или **1**, если его нужно искать через DHCP. Пользователь username – лицо, ответственное за данный компьютер, и он должен иметь на нем права запуска, просмотра и восстановления данных из резервных копий. Он же получит письмо от *BackupPC*, если что-то пойдет не так. Вы можете взвалить ответственность за управление пользовательскими резервными копиями на самих пользователей или оставить эту возможность себе. Можно добавить и четвертое поле – список других пользователей, которые могут пользоваться web-интерфейсом для данного хоста.

Большинство настроек задаются в файле **config.**Pl. Как это часто бывает, это огромный файл с подробными комментариями, но большинство настроек вначале можно оставить без изменений. Также нужно заметить, что этот файл и файлы настройки для каждого компьютера представляют собой скрипты Perl, поэтому будьте поосторожнее с их синтаксисом, если не знаете языка. В частности, пропуск точки с запятой в конце строки часто приводит к неожиданным результатам.

Разберем файл настройки

Первый раздел файла содержит основные настройки сервера, такие как **$Conf{MaxBackups}** — максимальное количество одновременных процессов созданий резервных копий, выполняемых сервером. Следующий раздел описывает, что и когда копировать, хотя эти параметры можно задать в файлах настройки отдельных компьютеров. *BackupPC* создает как полные, так и инкрементные резервные копии, и с параметрами **$Conf{FullPeriod}** и **$Conf{IncrPeriod}** в значениях по умолчанию это будет делаться раз в неделю и раз в день соответственно. На практике промежуток будет чуть длиннее, чем указанный здесь, поэтому установите чуть меньшие значения, чем целое число дней, в противном случае резервное копирование будет каждый раз выполняться чуть позже. При этом **$Conf{FullKeepCnt}** устанавливает число полных резервных копий, которые будут храниться. Установка его в **1** оставляет только самую свежую резервную копию, тогда как список чисел позволит вам гибко управлять этим процессом. Например, если установить значение **$Conf{FullKeepCnt} = [4, 0, 6]**, то будут храниться четыре последних резервных копии плюс по одной для каждого из предыдущих шести месяцев. Если вы не поняли, о чем я, прочтите комментарии в файле **config.pl**, там это объясняется подробнее.



› Web-интерфейс *BackupPC* неказист, но дело свое делает. Можно изменить общую конфигурацию, но не файлы, специфичные для данного хоста.

Теперь пора решить, когда выполнять резервное копирование. Если компьютеры работают постоянно, хорошее время для этого – ночь. В офисе, где они выключаются на несколько часов, подходяще время для этого – обед или, по крайней мере, время, когда они не заняты. **$Conf{BlackoutPeriods}** устанавливает время, когда резервные копии не делаются. Для задания нескольких интервалов можно воспользоваться массивом списков. Например:

```
$Conf{BlackoutPeriods} = [
{
hourBegin => 8.0,
hourEnd => 18.5,
weekDays => [1, 2, 3, 4, 5],
},
{
hourBegin => 8,
hourEnd => 13,
weekDays => [6],
},
];
```

Такая конфигурация вычеркивает интервалы с 8 утра до 6:30 вечера с понедельника по пятницу и с 8 утра до часа дня по субботам. Если время начала позже времени окончания, имеется в виду ночной интервал.

Впрочем, все это относится только к компьютерам, подключенным к сети постоянно или большую часть времени. Если компьютер был длительно отключен, *BackupPC* попробует создать »

Копирование сервера

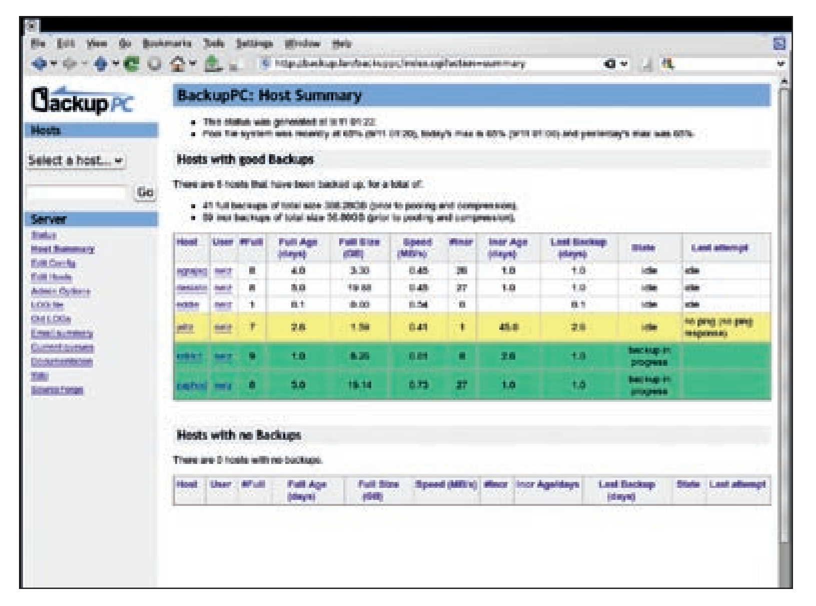
Если вы хотите, чтобы сервер сделал резервную копию самого себя, SSH – не самый эффективный вариант: гораздо лучше запустить *rsync* напрямую. Однако это следует делать от имени суперпользователя-root, поэтому воспользуйтесь **sudo**. Добавьте такие строки в начало файла настройки хоста:

```
$Conf{XferMethod} = 'rsync';
$Conf{RsyncClientCmd} = '/usr/bin/sudo $rsyncPath $argList+';
$Conf{RsyncClientRestoreCmd} = '/usr/bin/sudo $rsyncPath $argList+';
```

Затем запустите **visudo** в терминале и добавьте в конфигурацию **sudo** строку, чтобы разрешить пользователю *BackupPC* запускать *rsync* от имени root без запроса пароля:

```
backuppc ALL = NOPASSWD: /usr/bin/rsync
```

Позаботьтесь об удалении **/var/lib/BackupPC** из списка каталогов для резервирования, иначе место на диске кончится со свистом.



› **Список клиентов: два зеленых обрабатываются сейчас, желтый — предупреждение о том, что компьютер недоступен.**

резервную копию сразу же, как обнаружит подключение, независимо от вычеркнутых интервалов. Ведь может случиться так, что пользователь зашел лишь ненадолго, и система «ловит момент» поработать.

Закончив основную настройку, задайте параметры для отдельных компьютеров. Это делается в файле **/etc/BackupPC/pc/имя_хоста.pl**, где имя_хоста – имя хоста из файла **hosts**. Настройки для *rsync* и SMB различны, поэтому сначала рассмотрим *rsync*. В этом файле нужно указать три основных настройки – используемый протокол, каталоги, резервные копии которых нужно создать, и каталоги, которые нужно пропустить. Например:

```
$Conf{XferMethod} = 'rsync';
$Conf{RsyncShareName} = ['/'];
$Conf{BackupFilesExclude} = ['lost+found', '/var/run/*', '/var/tmp/*', '/var/lock/*', '/dev', '/proc/*', '/sys/*', '/media/*', '/tmp/*'];
```

Здесь для передачи файлов используется *rsync*, работающий через SSH, и это самый эффективный метод, так как передаются только изменения. **RsyncShareName** — это путь, резервную копию которого нужно создать (здесь – корень файловой системы, то есть все), и потом мы используем **BackupFilesExclude**, чтобы пропустить содержимое различных каталогов (но не сами каталоги, поэтому мы используем **/dirname/*** вместо **/dirname**). Незачем делать резервную копию временных файлов, содержимого сменных устройств, смонтированных на каталог **/media**, и, возможно, **/mnt** или содержимого виртуальных файловых систем в каталогах **/dev**, **/proc** и **/sys**.

Есть и другой способ исключить эти каталоги, с более четкой конфигурацией. Найдите параметр **$Conf{RsyncArgs}** в **config.pl** и добавьте **--one-file-system** в список аргументов. Эта настройка велит *rsync* не выходить за границы файловой системы, поэтому сам каталог **/** будет обработан, но ни одна файловая система, смонтированная на каталоги внутри него, не будет, и сменные устройства и виртуальные файловые системы исключатся автоматически. Обратите внимание, что если **/home** или **/var** смонтированы на отдельную файловую систему, потребуется отметить это в **RsyncShareName** следующим образом:

```
$Conf{RsyncShareName} = ['/', '/home',];
```

Нужно сделать еще кое-что. Так как *rsync* работает через SSH, нужно дать права доступа пользователю **backuppc** на сервере. Если вы копируете не только домашние каталоги, этому пользователю потребуется доступ с правами root для чтения системных файлов. Однако каждый раз вводить пароль или оставлять его в файле настройки неудобно и небезопасно, поэтому для аутентификации используются ключи SSH. Это означает, что нужно сгенерировать пару ключей для пользователя **backuppc** на сервере следующей командой:

```
su -s /bin/sh -c "ssh-keygen -t rsa" backuppc
```

Она запустит **ssh-keygen** под пользователем **backuppc**. Для него не задана оболочка по умолчанию, поэтому укажите ее через параметр **-s**. При создании пары ключей в **/var/lib/BackupPC/.ssh** примите настройки по умолчанию. Скопируйте публичную часть ключа на USB-брелок, вставьте его в компьютер, данные с которого будут копироваться, и введите команду:

```
cat /media/usbstick/id_rsa.pub >>/root/.ssh/authorized_keys
```

Если у пользователя root на целевом компьютере еще нет пары ключей SSH, запустите **ssh-keygen -t rsa** и на этой машине.

Последнее подтверждение

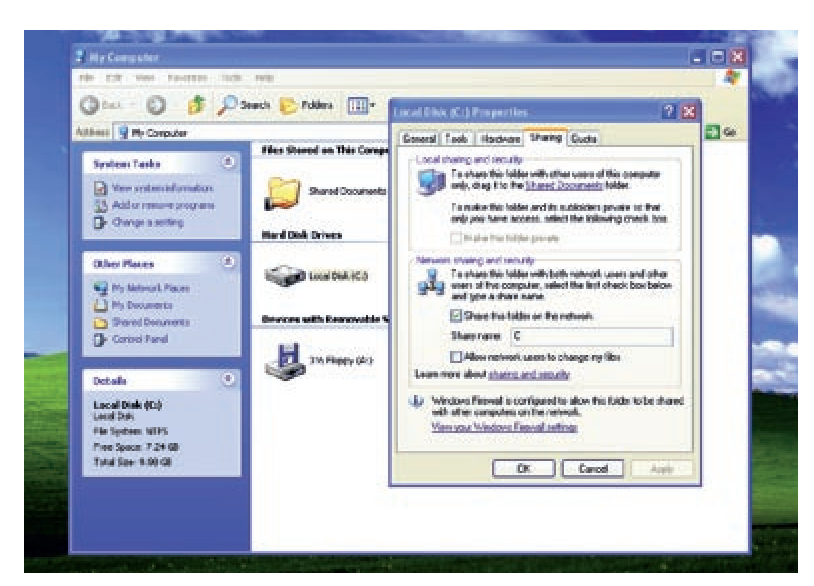
Есть еще один шаг, необходимый перед тем, как можно будет пользоваться SSH-соединением. При первом подключении к компьютеру, SSH перед записью аутентификационных данных в файл **known_hosts** запрашивает подтверждение. Это остановило бы автоматическое резервирование, поэтому зайдите на каждый компьютер вручную с помощью следующей команды:

```
su -s /bin/sh -c "ssh root@целевой_компьютер echo" backuppc
```

По сути дела, эта команда подключается, выводит пустую строку и разрывает соединение, но она будет запрашивать подтверждение при первом подключении к каждому хосту. Теперь всего готово для запуска *BackupPC* и создания резервных копий компьютеров Linux. Читайте дальше – и узнаете, как сделать то же самое с компьютерами Windows…

Для связи с клиентами Windows *BackupPC* использует SMB. В некоторых версиях Windows диск **C:** открыт для общего доступа как **C$** только для администраторов, в противном случае придется предоставить к нему общий доступ. Основной файл настройки для клиента Windows выглядит примерно так:

```
$Conf{XferMethod} = 'smb';
```



› **Если у вас несколько компьютеров с Windows, перед началом резервного копирования может потребоваться открытие общего доступа к диску C:.**

Зачем нужно Visudo?

При изменении настроек **sudo** воспользуйтесь **visudo** вместо непосредственного редактирования файла **/etc/sudoers**. Для правки файла будет использован редактор из переменной **$EDITOR**, не обязательно *Vi*, и перед перезаписью оригинала выполняется проверка. Это может спасти вас от самоблокировки из-за опечатки.

```
$Conf{SmbShareName} = ['C$'];
$Conf{SmbShareUserName} = 'backup';
$Conf{SmbSharePasswd} = 'yeahright';
```

Так как здесь указан пароль администратора, нужно гарантировать, что файл не прочесть никаким другим пользователям:

```
chown backuppc: hostname.pl
chmod 600 hostname.pl
```

Web-интерфейс

Ну вот, мы установили сервер и настроили его на резервное копирование данных с одного или нескольких клиентов; но совершенно не представляем, что и когда он скопировал, так как пользовательского интерфейса у нас нет. При установке сервера *BackupPC* также добавляется несколько скриптов Perl для управления программой через браузер. Их расположение зависит от дистрибутива и выбранного установочного пакета; у меня это каталог **/var/www/localhost/backuppc**. Чтобы воспользоваться интерфейсом, может потребоваться изменение конфигурации *Apache*. Во-первых, убедитесь, что опция **AllowOverride** для этого каталога установлена в **all**. Директива **AllowOverride** позволяет переопределять некоторые настройки каталога и его подкаталогов, подключая их из файла **.htaccess**. В зависимости от дистрибутива и установки, эта директива может находиться в файле **vhost**, по умолчанию в **/etc/apache**, или в основном файле настройки. Некоторые дистрибутивы сделают это автоматически, но если у вас есть сомнения, добавьте следующие строки в основной конфигурационный файл:

```
<Directory /var/www/localhost/backuppc>
AllowOverride all
</Directory>
```

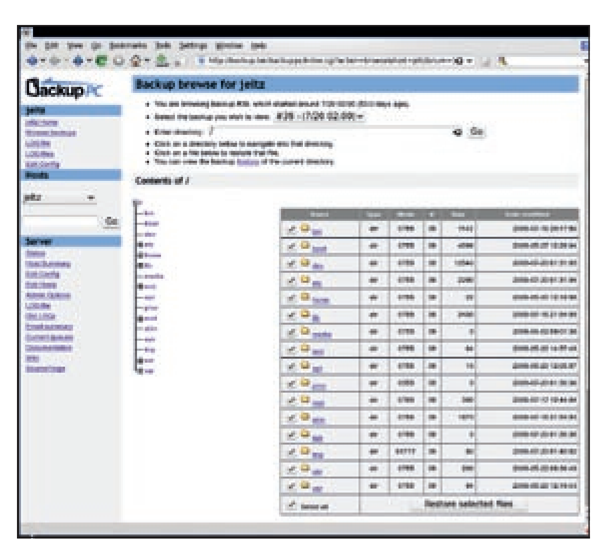
Пользователь **backuppc** будет создан с паролем по умолчанию или случайным паролем; измените его командой

```
htpasswd2 -c /etc/BackupPC/authUser имя_пользователя
```

Эта команда удалит существующего пользователя и пароль и запросит у вас новый пароль. Имя пользователя должно быть таким, какое установлено для **$Conf{CgiAdminUsers}** в файле **config.**Pl. Если в файле хоста есть другие пользователи, запустите команду для каждого из них, но не указывайте ключ **-c** для второго и последующих пользователей, иначе данные всех предыдущих пользователей будут стерты. Использовать для этого виртуальный хост или нет, решать вам, хотя запускать *BackupPC* в установке *Apache* для публичного web-сервера – не лучшая идея (см. врезку *Безопасность*). Впрочем, если компьютер – только сервер резервного копирования и не будет доступен извне, беспокоиться не о чем.

> «Можно мирно спать, зная, что BackupPC обо всем позаботится.»

Теперь можно открыть в браузере адрес http://имя_хоста/backuppc и войти с только что заданными именем и паролем. Web-интерфейс позволяет просматривать различные компьютеры и резервные копии, запускать операции резервного копирования или восстановления, читать документацию и даже менять конфигурационные файлы. Для начала зайдите на страницу Host Summary [Сводка хостов], кликните на первом компьютере и нажмите кнопку Start Full Backup [Запуск полного резервирования], чтобы начать резервное копирование. На странице Status [Состояние] будут показаны создаваемые резервные копии и информация обо всех ошибках, которые могут возникнуть. Если появляется ошибка «невозможно прочесть данные» ('unable to read') с компьютера Linux, проверьте, что можете подключиться к нему через SSH как пользователь **backuppc**. Восстановить отдельные файлы или целые файловые системы можно непосредственно на нужный компьютер или в tar- или zip-архив, а потом при желании скопировать его вручную.



› **Просто отметьте нужные файлы и нажмите Restore, чтобы вернуть их в состояние любой из сохраненных резервных копий.**

Безопасность

Для резервирования системных файлов потребуется дать пользователю **backuppc** доступ к клиентским компьютерам с правами root. Это означает, что нужно обязательно убедиться в том, что сервер и работающий с ним web-сервер нельзя взломать извне.

Если сервер резервного копирования работает не на отдельном компьютере и *Apache* обслуживает на нем другие web-приложения, есть шанс, что «дыра в безопасности» в них может дать хакеру доступ к вашим резервным копиям. Один из способов избежать этого – запустить отдельный экземпляр *Apache* для резервных копий. Для этого достаточно сделать копию каталога **/etc/apache**, удалить оттуда все ссылки на другие хосты и изменить порт в основном конфигурационном файле. Можно также задать в качестве корневого каталога [document root] web-каталог *BackupPC*, который нужно убрать подальше от существующего корневого каталога. Потом запустите второй экземпляр *Apache* командой:

```
apache2 -f /путь/к/основной/конфигурации
```

и подключите его к http://имя_хоста:номер_порта

Забирайте выше

Здесь мы лишь поверхностно коснулись возможностей *BackupPC*. Начальная установка довольно сложна, но как только, вам на радость, программа заработает, больше делать ничего не нужно. Чтобы добавить новые компьютеры для резервного копирования, обычно достаточно лишь дописать строку в файл **hosts** и создать новый файл **hostname.pl**, скопировав и изменив существующий. В базовой установке еще есть с чем повозиться – поройтесь в документации и комментариях файла настройки, коли охота. А можно просто оставить программу в этом состоянии и мирно спать с осознанием того, что *BackupPC* обо всем позаботится и отправит вам письмо, если что-то пойдет не так.

И последнее, что нужно знать о *BackupPC*. Здесь нет возможности проверять резервные копии, и сохранность ваших данных обеспечивается только проверками нижележащего протокола. Однако это не означает невозможности убедиться, что архив содержит то, что ему полагается. Для этого выберите все файлы в резервной копии и сохраните их в tar-архив, а затем проверьте его с помощью аргумента **--compare** команды tar. LXF

Lua: Встроим его

Часть 4: Вот уже четвертый номер мы пропагандируем Lua как хороший встраиваемый язык, и теперь пришла пора подтвердить слова делом. **Андрей Боровский** скрестит его с C и C++



Наш эксперт

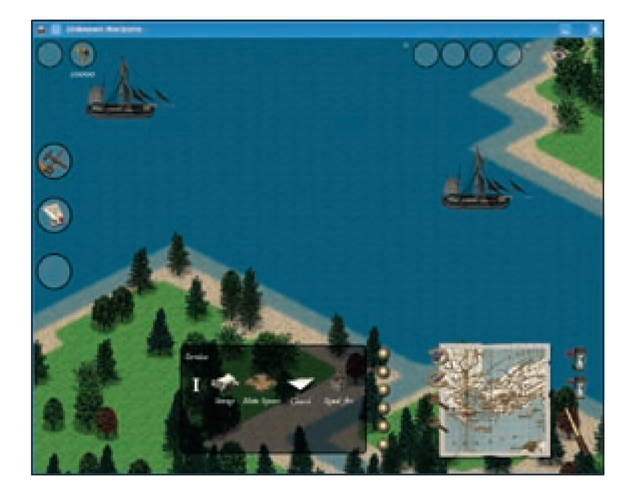
Андрей Боровский
За последние четыре года написал для **LXF** столько разных программ, что задумался о включении в них средств автоматизации.

Говорят, что когда Билл Гейтс и Пол Аллен писали первый интерпретатор Basic для «Альтаира», они не пользовались генераторами лексических и синтаксических анализаторов и формальными грамматиками, потому что не знали об их существовании. Тут можно усмотреть как проявление гениальности основателей Microsoft, так и отсутствие надлежащей квалификации. Как бы там ни было, современные программисты также могут наделять свои приложения встроенными интерпретаторами скриптовых языков, ничего не зная о принципах написания последних. Умные люди упростили решение этой задачи до такой степени, что теперь результатами их труда могут пользоваться все желающие. Остается только надеяться, что с годами избалованные программисты не станут глупее.

Все синтаксические красоты Lua вряд ли привлекли бы к нему внимание, если бы не главное: возможность встраивать интерпретатор в программы, написанные на C и других компилируемых языках. О том, что это дает разработчику, мы говорили в **LXF122**. С точки зрения программного интерфейса, объединение Lua и C позволяет решить две (как правило, связанные) задачи: управление программой Lua из приложения на C и вызов из программ Lua функций, написанных на C (в этом случае можно сказать, что программа Lua управляет C-кодом).

Состояние и стек

Ядро интерпретатора Lua находится в библиотеке *liblua*. Программа-интерпретатор, которой мы пользовались до сих пор (файл **lua5.1**), представляет собой лишь тоненькую обертку вокруг нее. Чтобы наделить программу на C возможностью использовать язык Lua, нужно связать ее с *liblua* и задействовать экспортируемый библиотекой API. Прежде чем выполнять примеры из этой статьи, убедитесь, что файл **liblua.so** в вашей системе является ссылкой на библиотеку **liblua.so.5.1**. API C довольно заметно изменился при переходе от версии Lua 5.0 к 5.1, а в наших примерах мы будем использовать API последней версии.



› **Игра UnknownHorizon — пример программы, использующей интерпретатор Lua.**

Два наиболее важных понятия в Lua C API – это «состояние Lua» (Lua state) и стек. Первое есть структура, содержащая сведения об интерпретаторе, к которым можно получить доступ из функции и макросов Lua API. Адрес данной структуры является обязательным параметром для всех функций и макросов Lua C API (и всегда передается в качестве первого аргумента, так что дальше мы не будем каждый раз о нем упоминать). Для тех, кто регулярно имеет дело с различными программными интерфейсами в стиле C (скажем, *GTK*), структура, описывающая состояние Lua, не представляет собой ничего необычного – ее аналоги, содержащие информацию о состоянии программируемого элемента (окна, кодека, драйвера устройства и тому подобного), есть практически везде. Указатель на структуру, описывающую состояние Lua, можно рассматривать как дескриптор экземпляра интерпретатора Lua (только нужно помнить, что экземпляра как такового не существует).

Если состояние Lua выглядит абстрактным понятием, то стек Lua еще более абстрактен. Как мы знаем, переменные Lua очень сильно отличаются от переменных C, поэтому нет возможности установить прямое соответствие между таковыми в программе-хозяине и сценарии Lua. Стек играет роль средства для обмена данными, и также является частью состояния Lua. В него можно помещать элементы, которые затем могут быть извлечены в обратном порядке. Практически все функции Lua C API выполняют операции над элементами стека и помещают в него результат. Поскольку переменные Lua полиморфны, ячейки стека могут содержать данные любого типа, поддерживаемого Lua. Прежде чем использовать данные из стека в программе на C, необходимо убедиться, что они имеют требуемый тип. С помощью стека между программой на C и кодом Lua передаются не только простые переменные, но и ссылки на функции и таблицы, описывающие загруженные модули. Вот почему стек Lua является важнейшей концепцией всего API.

Запутались? Все это не так сложно, как кажется. Рассмотрим пример программы на C, загружающий скрипт Lua (файл **runlua.c** на **LXF**DVD):

```
#include <lua.h>
#include <lauxlib.h>
#include <stdlib.h>
#include <stdio.h>
int main(void)
{
    int result;
    lua_State * L;
    L = luaL_newstate();
    luaL_openlibs(L);
    result = luaL_loadfile(L, "script.lua");
    if (result) {
```

» **Месяц назад** Мы научились использовать модули Lua и писать собственные.

в код




```
        fprintf(stderr, "Ошибка загрузки: %s\n", lua_tostring(L, -1));
        exit(1);
    }
    lua_pushnumber(L, 2);
    lua_setglobal(L, "var");
    result = lua_pcall(L, 0, LUA_MULTRET, 0);
    if (result) {
        fprintf(stderr, "Ошибка выполнения: %s\n", lua_tostring(L,
-1));
        exit(1);
    }
    printf("Успешное завершение\n");
    lua_close(L);
    return EXIT_SUCCESS;
}
```

Здесь считывается файл **script.lua**, создается глобальная переменная Lua **var** со значением 2 и выполняется Lua-программа. Элементы Lua C API описаны в заголовочных файлах **lua.h** и **lauxlib.h**. Переменная L – это указатель на структуру, описывающую состояние Lua. Саму структуру мы создаем с помощью функции **luaL_newstate()**. Функция **luaL_openlibs()** открывает стандартные библиотеки Lua для заданного состояния. Если ее не вызывать, загруженная программа Lua не сможет обращаться к элементам стандартных библиотек Lua, например, функции **print()**. Вызов **luaL_loadfile()** загружает текст программы Lua из файла. В случае успеха функция возвращает 0; ненулевое значение свидетельствует об ошибке.

> «Библиотека Lua возвращает подробные описания ошибок.»

Библиотека Lua возвращает подробные текстовые описания всех обнаруженных ошибок, причем использует для этого стандартный механизм передачи данных между Lua и программой-хозяином, то есть стек. В случае возникновения ошибки мы читаем ее описание посредством функции **lua_tostring()**. Обратите внимание на ее второй аргумент. Как и в большинстве функций Lua API, имеющих дело с данными Lua, в качестве источника сведений для **lua_tostring()** мы указываем ячейку стека. Они обозначаются индексами, причем начиная с единицы, а не с нуля. Вызов функции

```
lua_tostring(L, -1)
```

преобразует в строку C значение, находящееся в верхней ячейке стека (т.е. попавшее в стек последним). Аналогично, конструкция

```
lua_tostring(L, 1)
```

считывает значение, которое было помещено в стек первым. Ячейку, где находится самое последнее из помещенных в стек значений (и на которую ссылается индекс -1), мы называем вершиной стека. Ячейку, где находится самое первое помещенное в стек значение (самое старое, которому соответствует индекс 1), мы называем дном. Документация Lua придерживается противоположных обозначений (то, что мы называем дном стека, там считается вершиной, и наоборот). В данном примере будет считано описание ошибки, которое помещается на вершину стека.

Важно отметить, что функция **lua_tostring()** (а также **lua_tonumber()** и ей подобные) не удаляет значение из стека. При работе с функциями Lua API следует всегда помнить, как они влияют на стек. Библиотеки Lua не следят за переполнением стека – вы сами должны заботиться о том, чтобы в нем не было ничего лишнего. Для удаления из стека заданного значения предназначена функция **lua_pop()**. Мы не вызываем ее после **lua_tostring()** только потому, что программа в этом случае все равно завершится и очищать стек специально нет необходимости.

Если текст программы загружен успешно, функция **luaL_loadfile()** создает выполняемый интерпретатором Lua фрагмент и помещает его на вершину стека. Далее мы создаем глобальную переменную **var**. Функция **lua_pushnumber()** помещает в стек числовое значение. Макрос **lua_setglobal()** создает глобальную переменную и присваивает ей значение из ячейки, находящейся на вершине стека. Если мы посмотрим определение макроса **lua_setglobal()**, то увидим, что он сводится к вызову **lua_setfield()**. Первым аргументом этой функции является указатель на структуру, описывающую состояние Lua, вторым – индекс таблицы, в которую добавляется переменная, а третьим – строка C с ее именем. Позволю себе напомнить, что все переменные Lua являются элементами каких-либо таблиц. Так, глобальные переменные содержатся в **_G** (**LXF124**) – именно с этой таблицей и работает макрос **lua_setglobal()**. Разумеется, с помощью функции **lua_setfield()** (и макроса **lua_setglobal()**) можно не только создавать новые переменные, но и модифицировать значения уже существующих. С помощью этих же функций можно уничтожать переменные Lua. Делается это так же, как и внутри Lua-кода, то есть путем присвоения переменной значения nil. Для записи его в стек предназначена специальная функция **lua_pushnil()**. Функция **lua_setfield()** удаляет значение с вершины стека, так что вызов **lua_pop()** после нее не нужен.

Функция **lua_pcall()** выполняет самую волшебную часть нашей программы – запускает сценарий Lua на выполнение. Второй ее аргумент – число параметров, переданных выполняемому фрагменту (в нашем примере – 0), третий – число значений, которые возвращает фрагмент (мы используем константу **LUA_MULTRET**, указывающую, что оно может быть переменным).

Помимо функции **lua_pcall()**, API предоставляет нам **lua_call()**. Те, кто читал предыдущую статью, наверняка уже догадались, в чем разница между ними. Функция **lua_pcall()** выполняет загруженный код в «защищенном режиме», то есть подавляет все возникшие во время выполнения кода Lua ошибки, а не передает их на более высокий уровень. Но откуда функция **lua_pcall()** знает, какой фрагмент кода Lua она должна выполнить? В процессе ее вызова из стека извлекаются аргументы, переданные фрагменту кода Lua (если они есть), а после них **lua_pcall()** ожидает увидеть ссылку на сам фрагмент. Она также извлекается из стека, и на ее месте оказываются значения, возвращенные выполненным фрагментом. Последний аргумент **lua_pcall()** – индекс в стеке функции Lua, используемой для обработки ошибок. Если он равен 0, то в случае возникновения ошибки на вершине стека оказывается текстовое описание, которое мы выводим так же, как и в случае с **luaL_loadfile()**. Теперь нетрудно понять, как интерпретатор Lua выполняет несколько фрагментов программы, содержащихся в разных файлах, в едином контексте. Все, что для этого нужно – вызвать несколько функций **lua_[p]call()** с одной и той же пе- »

ременной, описывающей состояние интерпретатора Lua. Наконец, мы уничтожаем структуру, описывающую состояние Lua, функцией **lua_close()**.

Файл **script.lua**, который мы загружаем на выполнение в программе-примере, может содержать любой корректный фрагмент программы Lua. Нашу среду выполнения Lua отличает то, что фрагменту доступна глобальная переменная **var**. Это можно проверить с помощью простейшей конструкции:

```
print("переменная var", var)
```

Сохранив эту строку в файле **script.lua**, скомпилируйте нашу С-программу:

```
gcc runlua.c -o runlua -llua
```

Наберите **./runlua** и убедитесь, что приложение работает. Попробуйте отредактировать текст **script.lua** и посмотрите, как наша программа реагирует на различные ошибки в коде Lua.

И снова калькулятор

В прошлый раз мы использовали возможности Lua для создания программы-калькулятора. Теперь давайте попробуем применить Lua для добавления функций калькулятора в приложение C++ (файл **calc.cpp**):

```
#include <iostream>
#include <string>
extern "C" {
#include "lualib.h"
#include "lauxlib.h"
}
using namespace std;
size_t l;
const char * reader (lua_State *L, void *data, size_t *size) {
  char * result;
  * size = l;
  if (!l)
    result = 0;
  else
    result = (char *) data;
  l = 0;
  return result;
}
int main()
{
  int result;
  lua_State * L;
  L = luaL_newstate();
  luaL_openlibs(L);
  while (true) {
    cout << "Введите строку или нажмите Ctrl-C" << endl;
    string s;
    cin >> s;
    s = "do return " + s + " end \0";
    l = s.length() + 1;
    result = lua_load(L, reader, &s[0], "calc");
    if (result) {
      cout << lua_tostring(L, -1) << endl;
    }
    else {
      result = lua_pcall(L, 0, 1, 0);
      if (result != LUA_ERRRUN)
        cout << lua_tonumber(L, -1) << endl;
      else
        cout << lua_tostring(L, -1) << endl;
    }
    lua_pop(L, 1);
  }
  lua_close(L);
}
```

› Наш калькулятор может вычислить все, даже тангенс от π/2!

Программа считывает строку со стандартного потока ввода и пытается вычислить содержащееся в ней математическое выражение (а на самом деле – все, что догадался набрать пользователь, так что будьте осторожны с этим в реальных приложениях!). Наш микрокалькулятор принимает выражения вида **2*(3+4)**, **math.sin(3.14)** и им подобные. Работа выполняется в бесконечном цикле, из которого можно выйти с помощью Ctrl+C.

Обратите внимание, что в программе, написанной на C++, нужно явным образом указать, что функции Lua API экспортируются в формате C. После того как мы считали строку (переменная **s**), нужно скомпилировать ее в функцию Lua. В калькуляторе, написанном на чистом Lua, для этого можно было использовать специальные функции. В Lua C API такие тоже были (например, **lua_dostring()**), но теперь они признаны устаревшими. Вместо них следует использовать **lua_load()**, которая умеет загружать исходный текст программы Lua из любого источника с помощью специальной вспомогательной функции. В нашем примере это **reader()**. При каждом вызове она должна вернуть указатель на новый фрагмент исходного текста программы Lua. Длина очередного фрагмента возвращается в параметре **size**. Параметр **data** представляет собой указатель на данные, определенные программистом. Функция **reader()** сигнализирует о том, что она прочитала весь исходный текст, возвращая значение NULL. При первом вызове наша функция просто преобразует введенную пользователем программы строку C++ в **char *** и возвращает ее длину. При втором вызове **reader()** сразу же возвращает NULL.

Вернемся к **lua_load()**. Ее второй параметр – адрес вспомогательной функции для чтения исходного текста, третий – адрес данных для вспомогательной функции. Далее следует строка с именем загружаемого фрагмента (можно передать пустую строку). Обработка ошибок, возникших во время выполнения функции **lua_load()**, осуществляется так же, как и в случае с **luaL_loadfile()**. Аналогично, функция **lua_load()** создает выполняемый интерпретатором Lua фрагмент и помещает его на вершину стека. После завершения **lua_pcall()** стек содержит значение, вычисленное нашим калькулятором. В случае ошибки оно будет иметь тип **nil**, и мы считываем его с верхушки стека с помощью вызова **lua_tonumber()**. Если вершина стека содержит нечисловое значение, вызов **lua_tonumber()** возвращает 0. После этого нам остается только удалить значение из стека с помощью **lua_pop()**.

И наоборот

Посмотрим теперь, как вызвать из Lua функцию, написанную на C. Разумеется, в Lua нельзя импортировать любую функцию из C-библиотеки – слишком уж разные это языки программирования. Тем не менее, написать C-функцию в «экспортном варианте» оказывается на удивление просто. Рассмотрим сначала исходный текст библиотеки *testlib* (файл **testlib.c**), написанной на C:

```
#include "lua.h"
```

```
andrei@linux-3nc6:~/Articles/lua-4
andrei@linux-3nc6:~/Articles/lua-4> cat ltestlib.lua
require "testlib"

print("Calling C function")
res = testlib.system("/usr/bin/mc")
if res ~= 0 then
print("error", res)
else
print("normal exit")
endandrei@linux-3nc6:~/Articles/lua-4>
andrei@linux-3nc6:~/Articles/lua-4>
andrei@linux-3nc6:~/Articles/lua-4>
andrei@linux-3nc6:~/Articles/lua-4> cat ltestlib.lua
require "testlib"

print("Calling C function")
res = testlib.system("/usr/bin/mc")
if res ~= 0 then
print("error", res)
else
print("normal exit")
andrei@linux-3nc6:~/Articles/lua-4> []
```

› Экспортируемая из C функция вызывается в Lua как любая другая.

```
#include "lualib.h"
#include "lauxlib.h"
#include <stdlib.h>
static int _system (lua_State *L)
{
  int result = system(lua_tostring(L, -1));
  lua_pushinteger(L, result);
  return 1;
}
int luaopen_testlib(lua_State *L)
{
  static const luaL_reg Map [] = {{"system", _system},
{NULL,NULL}} ;
  luaL_register(L, "testlib", Map);
  return 1;
}
```

Данная библиотека предоставляет Lua одну функцию – **system()**. Функция **testlib.system()** (под этим именем она будет доступна в Lua) делает то же, что и ее тезка из стандартной библиотеки C (разумеется, в стандартной библиотеке Lua она тоже есть). Во избежание неоднозначностей, в нашей C-библиотеке экспортируемой функции присвоено имя **_system()**.

Чтобы понять, как работает экспорт C-функций, необходимо различать передачу аргументов и возврат значений в контексте C и те же действия в контексте Lua. Единственным аргументом экспортируемой функции в контексте C должен быть указатель на структуру **lua_State**, а возвращаемым значением (в контексте C) – число значений, возвращаемых в контексте **Lua**. Да, вы правильно поняли: в контексте Lua экспортируемая функция может получать произвольное количество аргументов различных типов и возвращать любое число значений. Все аргументы и значения в контексте Lua передаются, разумеется, через стек. Мы считываем единственный аргумент функции **testlib.system()** с вершины стека, превращаем его в строку, вызываем C-функцию **system()** и помещаем результат выполнения функции в стек Lua. Поскольку в контексте Lua наша функция всегда возвращает только одно значение, в C-коде мы пишем **return 1**.

Чтобы подготовить библиотеку к работе с Lua, нам надо написать еще одну функцию – **luaopen_libname()**, где **libname** соответствует имени библиотеки. Она будет вызвана интерпретатором Lua при загрузке нашей библиотеки. Именно она позволяет использовать C-библиотеку как стандартный пакет Lua.

Массив **Map** состоит из пар «имя функции в контексте Lua – указатель на C-функцию». Эти данные интерпретатор Lua будет использовать для вызова функции, написанной на C. Между прочим, поскольку имя функции в библиотеке C не имеет значения для интерпретатора, функции, в принципе, можно экспортировать и из кода C++, не указывая формат (это не касается **luaopen_*()**, которую интерпретатор ищет по имени C). Функция **luaL_register()** регистрирует новую библиотеку и помещает ссылку на соответствующий объект на вершину стека. Вторым аргументом функции **luaL_register()** должно быть имя библиотеки в контексте Lua, которое может и не совпадать с именем разделяемого модуля.

Для компиляции библиотеки скомандуем

```
gcc testlib.c -shared -o testlib.so
```

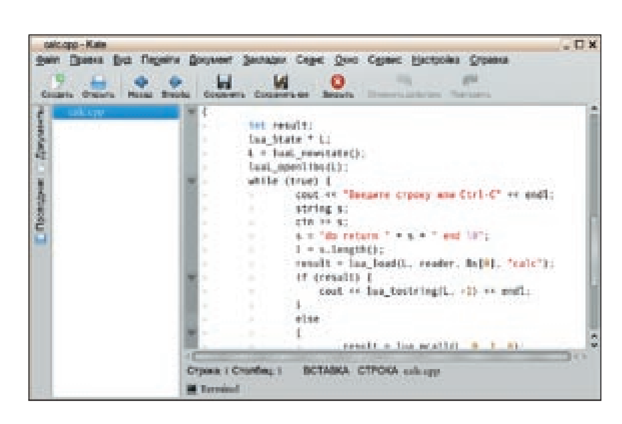
Теперь можно перейти к Lua-коду:

```
require "testlib"
print("Вызов функции C")
res = testlib.system("/usr/bin/mc")
if res ~= 0 then
print("Код завершения программы", res)
else
print("Нормальный выход")
end
```

Наш модуль *testlib* подключается к программе Lua так же, как и обычные библиотеки Lua (**LXF124**). Обратите внимание, что хотя в библиотеке *testlib* и используются функции Lua API, при компиляции мы не связываем ее с *liblua*. В этом нет необходимости, так как библиотека *testlib* будет загружена интерпретатором в то же адресное пространство, что и *liblua*. Между прочим, экспортировать функции C в Lua можно не только из разделяемых библиотек, но и непосредственно из файла программы, загружающей код Lua на выполнение. В этом случае нам пригодится другая функция Lua API – **lua_register()**.

Каким образом функция C может узнать, сколько аргументов ей передано? В этом ей поможет функция **lua_gettop()**, которая возвращает число элементов в стеке. В момент вызова функции C оно равно количеству аргументов, переданных функции. Строго говоря, **lua_gettop()** возвращает индекс последней ячейки стека, которая в терминологии Lua именуется вершиной (в описании стека я, как и авторы многих руководств по Lua, придерживался других обозначений: вершина – первый элемент стека). Этой функцией можно пользоваться и для того, чтобы узнать, не переполнен ли стек. Глубина стека Lua контролируется константой **MAX_STACK**, значение которой невелико (обычно не больше 32), но для любого разумного использования этого должно быть достаточно. В конце концов, глубина стека математического сопроцессора у Intel составляет всего 8 ячеек.

Встраивание скриптовых языков программирования в пользовательские программы вполне соответствует идеологии Unix, согласно которой все, что может быть настроено пользователем под свои нужды, должно быть не зашито в исходном коде программы, а вынесено в отдельные системы конфигурации. Язык Lua – простое и мощное средство, позволяющее наделить вашу программу такими возможностями. LXF



› Универсальные редакторы, типа *Kate*, можно использовать и для C++, и для Lua.

Python: Сделаем

Конвертирование документов по запросу при помощи монстра а-ля доктор Моро – гибрида Google Docs, **Ника Вейча**, Python и Fuse.





Наш эксперт

Ник Вейч выяснил, что приложения для распознавания голоса не годятся для диктовки кода. Скобка закрывается.

Поработайте с Unix-системами подольше, и все для вас станет файлом. Ваш список задач? Это файл. Дата? Тоже файл. Ваша мышь? И она – файл. Пластиковый пакет, куда вы вкладываете бумаги – ну, вы в курсе.

Да, абстрактное восприятие всего в виде файлов имеет смысл в среде со множеством инструментов для использования файлов и работы с ними. Их можно копировать, читать, выводить список и перемещать при помощи нескольких команд. А каков будет мир, если мы сделаем файлом Интернет? Давайте посмотрим...

Применим Fuse

Fuse, или Filesystem in Userspace [файловая система пространства пользователя] – это проект, стартовавший в 2004 г. Его цель – предоставить способ монтирования файловых систем на уровне пользователя и сделать возможным написание программных виртуальных файловых систем. Работает он прекрасно, и, что замечательно, для создания Fuse-приложений нужен всего лишь Fuse-модуль ядра и библиотеки для любимого языка программирования. На данном уроке мы приложим руки к модулю Python под названием *python-fuse*.

Создать виртуальную файловую систему не просто, но и не так сложно, как думают многие. Большую часть работы выполняет Fuse, а вам остается только предоставить код, который решает, что содержит файловая система, и возвращает информацию стандартным вызовам ОС для управления файлами. Вам не стоит беспокоиться о способах выполнения копирования – нужны только низкоуровневые вещи: имя файла, что в нем находится и какие атрибуты у него в файловой системе.

Простая реализация Fuse не требует много кода, если ваш класс – потомок класса fuse Python: все что нужно сделать – это определить несколько методов, зависящих от того, какие возможности вы хотите реализовать в файловой системе. В простейшем случае реализуйте следующее:

- **getattr** возвращает такие значения, как размер файла и права.
- **readdir** возвращает список содержимого каталога.
- **open** возвращает дескриптор открытого файла.
- **read** возвращает содержимое файла.

Для файловой системы с возможностью записи следует описать дополнительные методы. Вот как может выглядеть простая реализация:

```
class MyStat(fuse.Stat):
  def __init__(self):
    self.st_mode = 0
    self.st_ino = 0
    self.st_dev = 0
    self.st_nlink = 0
    self.st_uid = 0
    self.st_gid = 0
    self.st_size = 0
    self.st_atime = 0
    self.st_mtime = 0
    self.st_ctime = 0
class MyFS(Fuse):
  def getattr(self, path):
    st = MyStat()
    st.st_atime = int(time.time())
    st.st_mtime = st.st_atime
    st.st_ctime = st.st_atime
    st.st_mode = stat.S_IFDIR | 0755
    st.st_nlink = 2
    st.st_size=4096
    return st
  def readdir(self, path, offset):
    dirents =['.', '..']
    if path == '/':
      dirents.extend(['foo', 'bar'])
    for e in dirents:
      yield fuse.Direntry(e)
  def open(self, path, flags):
    filehandle=open(ваш_файл,"r")
    return filehandle
```

» **Месяц назад** Мы выложили в сеть фотки графическим загрузчиком изображений.

Google диском

```
def read(self, path, size, offset=0, filehandle=None):
    filehandle.seek(offset)
    buffer= filehandle.read(size)
    return buffer
```

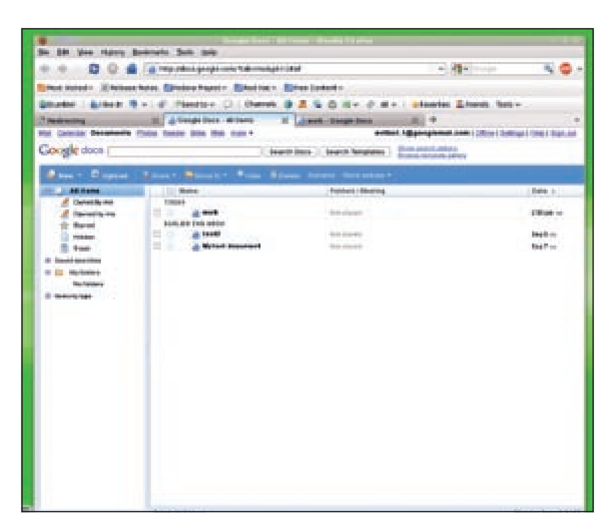
Эта (не рабочая) реализация показывает грубую структуру приложения, которое мы хотим создать. Стоит отметить несколько важных пунктов: каждый каталог должен содержать стандартные для Unix записи '.' и '..', и их всегда надо возвращать. Метод **open** может возвращать объект – обычно дескриптор файла, но мы сами вольны решать, чем он будет и как его использовать. Позже такой объект будет передан на вход метода **read**, что удобно, так как мы должны откуда-то брать данные, возвращаемые в ответ на его вызов (т. е. выполнять операции чтения).

На данный момент у нас упущена обработка ошибок. Она важнее, чем может показаться, потому что к любой смонтированной файловой системе современного дистрибутива обращается много стандартных служб ОС, выполняющих проверку наличия файла автозапуска, индексирование документов или очистку корзины. В финальной версии кода необходимо возвращать ошибки типа Файл не найден, для поддержания системы в рабочем состоянии.

Итак, мы собираемся расширить имеющееся, заставив файловую систему читать Google Docs. Наша цель – возможность монтирования файловой системы, отображающей наши документы и позволяющей читать их, словно они находятся на локальном диске.

Погружаемся в Docs

Для своего сервиса Docs, Google предоставляет API (**LXF120**); имеется также исчерпывающий набор официальных API-модулей для различных языков, включая Python. Он доступен во всех ведущих дистрибутивах, и вы можете установить *python-gdata* из вашего менеджера пакетов.



› **Экспортировать документы в различные форматы для работы на настольном компьютере неудобно, так почему бы не примонтировать Docs как диск?**

Перенаправление stderr

Запустив кусок кода вроде файловой системы Fuse, вы заметите, что он выполняется в собственном потоке. Это прекрасно, но означает, что вы не сможете получить из Python сообщения **stderr**, необходимые для отладки. Как известно, благодаря способу обработки исключений в коде Fuse, он может генерировать до 50 исключений в секунду, и никто о них не узнает. Ну, разве что файловая система не будет работать.

Вы можете создать файл журнала для захвата этих данных при помощи

```
fsock=open('/home/evilnick/error.log','a')
fsock.writelines("**Started**")
sys.stderr = fsock
```

Сам пакет *gdata* содержит API для практически всех сервисов Google и часто обновляется. Мы настоятельно рекомендуем обновить ваш пакет: в противном случае вы можете обнаружить, что результаты выполнения кода могут слегка отличаться.

Модуль *gdata* – это однородная масса, но пространство имен разбито на части, связанные с различными службами, и вам нет нужды работать со всей библиотекой: достаточно малой толики. Давайте опробуем что-нибудь в интерактивной оболочке Python:

```
>>> import gdata.docs.service
>>> user = '<ваше_имя_пользователм>@gmail.com'
>>> password = '<ваш_пароль>'
>>> client=gdata.docs.service.DocsService()
>>> r = client.ClientLogin(user,password, source="evil script")
>>> gdata.docs.service.SUPPORTED_FILETYPES
{'XLSX': 'application/vnd.openxmlf...
...
...'XLS': 'application/vnd.ms-excel', 'PNG': 'image/png'}
```

Как видите, мы просто импортировали кусок модуля и авторизовались в сервисе стандартным способом, путем создания объекта и вызова метода **ClientLogin**. Здесь указываются имя пользователя и пароль, а также строка источника, что упрощает для Google отслеживание действий клиента.

Google Docs поддерживает только документы в определенных форматах, что и понятно – это не файлообменник, и серверы Google должны иметь возможность понять, что к ним загружают. Вы можете найти текущий список путем вывода значения переменной модуля **SUPPORTED_FILETYPES** (в коде приведена не полностью), организованной в виде списка пар расширения файла и типов MIME.

Чтобы работать с сохраненными документам в Google Docs, необходимо знать, что они собой представляют. Объект **client** имеет метод для получения списка документов, возвращающий объект **feed**, содержащий пронумерованный список записей. Каждая запись имеет несколько свойств, включая имя, тип и уникальный ID-номер.

```
>>> list=client.GetDocumentListFeed()
>>> list
<gdata.docs.DocumentListFeed object at 0xa066f2c>
>>> for entry in list.entry:
... print (entry.title.text, entry.GetDocumentType, entry.resourceId.
text)
```

»

```
...("fun things we've done", 'document','document:dgg88xxxxxx
xfs')
('letters raw text', 'document', 'document:df4m88888888f9')
('cover', 'document', 'document:df4mf7888888888fk')
("Pamela's Cover Letter - June 2009", 'document',
'document:df4m88888888888gs')
('Untitled Presentation', 'presentation',
'presentation:dd26888888888d9')
('sharing and workflow', 'document', 'document:df4mf7c7_18888
888888g6')
```

Теперь мы можем воткнуть это в нашу конструкцию Fuse, чтобы получить список имеющихся документов. Мы можем пожелать загрузить их в различных форматах, поддерживаемых Google (DOC, ODT, PDF и так далее), так что вы можете создать каталоги и представить файлы в них, словно они имеют соответствующий тип:

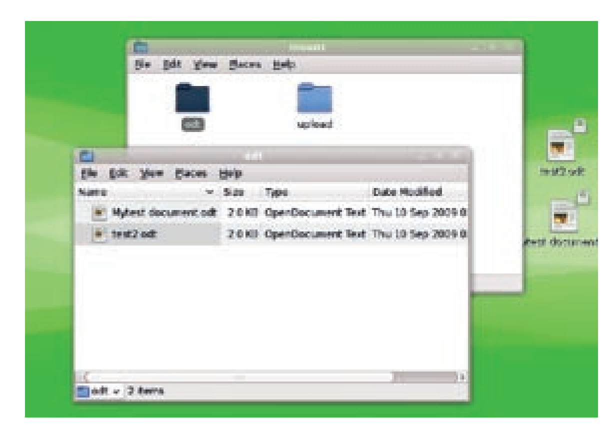
```
def readdir(self, path, offset):
  dirents =['.', '..']
  if path == '/':
    dirents.extend(['odt','doc','pdf'])
  if path == '/odt':
    # получаем документ с gmail
    client=gdata.docs.service.DocsService()
    client.ClientLogin(user, password, source="evil script")
    list=client.GetDocumentListFeed()
    for entry in list.entry:
      if entry.GetDocumentType() == 'document' :
        dirents.append(entry.title.text+'.odt')
    #вставьте код для других каталогов
  for e in dirents:
    yield fuse.Direntry(e)
```

Здесь мы включили код в класс нашей файловой системы. Теперь у нас есть корневой каталог – все относительно точки монтирования – и мы хотим вывести список содержимого наших каталогов с документами. Для первого из них, каталога **odt**, мы подключаемся к Google и загружаем список доступных документов, отсекая все, что не является ODF-документом. Далее используем метод fuse для возвращения путей с именами.

Это было вполне очевидно, но как только ОС выведет список каталога, она попробует проверить атрибуты каждой записи, поэтому нам необходимо создать и этот код. Для каталогов будем считать, что времена создания и изменения не важны. Мы можем создать глобальную переменную и помечать все при создании файловой системы, но это не стоит трудов. При выводе списка в оболочке каталоги всегда имеют размер 4096, и мы можем просто так и записать. Что касается прав, мы воспользуемся переменными модуля **stat**, созданного в рассмотренном выше простейшем примере, для установки стандартной записи для каталога и придания ей основных прав доступа.

› После монтирования файловой системы в каталоге появится список документов, в соответствии с типами используемых файлов. Учтите, что один и тот же документ может появиться во многих каталогах.

```
[evilnick@parmenion simplefuse]$ fusermount -u mount
[evilnick@parmenion simplefuse]$ python oldfuse.py mount
[evilnick@parmenion simplefuse]$ ls mount
odt  upload
[evilnick@parmenion simplefuse]$ ls -al mount/odt
total 8
drwxr-xr-x. 2 root root    0 2009-09-10 14:52 .
drwxr-xr-x. 2 root root 4096 2009-09-10 14:52 ..
-r--r--r--. 1 root root 2048 2009-09-10 14:52 Mytest document.odt
-r--r--r--. 1 root root 2048 2009-09-10 14:52 test2.odt
[evilnick@parmenion simplefuse]$ rm -f test2.odt
[evilnick@parmenion simplefuse]$ cp mount/odt/test2.odt .
[evilnick@parmenion simplefuse]$ ls -al test2.odt
-r--r--r--. 1 evilnick evilnick 3197 2009-09-10 14:53 test2.odt
[evilnick@parmenion simplefuse]$
```



› Настольный Linux может работать и с такими дисками. Gnome распознает его как устройство и даст вам после монтирования ярлык для него. Теперь копируйте файлы перетаскиванием.

С файлом все малость сложнее. Мы вполне можем создать все значения для прав, времени и так далее, но как быть с размером? Реальных способов узнать размер до завершения загрузки нет, а поскольку его не существует, то это невозможно. К счастью, мы часто можем вернуть стандартный размер файла, что позволит избежать множества проблем (хотя и приведет к неверному результату выполнения команды **du**).

Выдача неверного размера файла при запросе содержимого каталога – сравнительно безобидная шалость, но передача неверного размера чему-то, что пытается прочесть и открыть его целиком, может открыть ящик Пандоры: создастся дыра в пространственно-временном континууме, погибнет галактика, или открытие файла завершится на полполовине. И вновь есть уловка, которая поможет обойти это ограничение.

Идем на хитрость

Для начала, вам необходимо знать, что когда файловая система читает файл – например, при операции копирования – это всегда выполняется вызовами **open**, **getattr**, **read**, именно в таком порядке. Значит, нам надо только загрузить файл при получении вызова **open**, а затем поместить его в кэш. Вызов **getattr** можно использовать для проверки, что файл уже в кэше, и определения его точного размера. Поскольку файл проверяется до того, как начнется чтение, будет получен верный размер, и вызывающее нас приложение даже не будет знать, что мы его одурачили.

Более того, настройка кэша выполняется просто, если импортировать *tempfile*, стандартный модуль Python. Среди множества его опций есть метод создания временного каталога, который обычно размещается в **/tmp**, но это может меняться в зависимости от настроек системы. Метод, приведенный ниже, возвращает путь, и вы можете просто добавить, что нужно, до и после него:

```
cachedir=tempfile.mkdtemp(prefix='fuse-')
os.mkdir(os.path.join(cachedir, 'odt'))
os.mkdir(os.path.join(cachedir, 'pdf'))
```

Мы также создали несколько каталогов для хранения наших файлов; обратите внимание, что мы хотим воссоздать структуру нашей виртуальной файловой системы, чтобы легко находить кэшированные объекты.

После настройки каталога кэша его необходимо заполнять по мере надобности. Загрузка всего при каждом выводе списка каталога будет перебором, но нам необходимо скачивать последнюю версию файла при каждом его открытии. Поэтому при вызове **open** мы загружаем файл с Google и помещаем его в кэш. Далее нам необходимо открыть файл и вернуть дескриптор файла для нужд того, что его открывает.

```
def open(self, path, flags):
  path_element = path.split('/')
  fname=path_element[-1]
  #получаем файл с Google
  client=gdata.docs.service.DocsService()
  client.ClientLogin(user,password, source="evil script")
  q = gdata.docs.service.DocumentQuery()
  q['title'] = fname[:-4]
  q['title-exact'] = 'true'
  list = client.Query(q.ToUri())
  tfile=os.path.join(cachedir, path[1:])
  client.Export(list.entry[0], tfile)
  filehandle=open(tfile,"r")
  return filehandle
```

В этом коде мы используем один метод для загрузки файла с Google. Все, что нам известно о файле – его имя, и мы можем сделать запрос на его основе и направить его в Google, чтобы получить список файлов с указанными именами. Запрос отсылается, и мы получаем первый элемент списка, но если у вас два файла с одинаковыми именами, то это может стать проблемой. Одним из решений будет чтение всех документов в класс и сохранение дескрипторов на каждый файл, но при наличии файлов с одинаковыми именами в списке содержимого каталога все равно будет путаница. Мы будем считать, что все файлы имеют разные имена.

Экспорт неопределенности

Метод Export модуля Google весьма любопытен. Он принимает только объект **entry** – вы получаете его из запроса – и имя файла. Однако документы подчиняются некому принципу неопределенности, поскольку вы не знаете, в каком они формате, пока не спросите. То, как с этим справляется API, очень занятно: вместо указания желаемого типа файла он угадывается по имени, который вы передаете методу клиента. Другим моментом является то, что вместо передачи вам данных, клиент желает писать их в саму локальную файловую систему.

Теперь, имея работающий кэш, мы можем завершить уловку с атрибутами файла. Воспользуемся некоторыми выражениями-условиями, чтобы суметь вывести и атрибуты содержимого корневого каталога, и файлы в каталоге с нашими документами. Мы специально сделали блок условий для записей, которые появляются в нашем каталоге **/odt**, но, в предположении плоской структуры каталогов, вы можете легко расширить этот сервисный блок на все подкаталоги, изменив условия на проверку пути их трех элементов. Мы также возвращаем корректный код ошибки о несуществующих элементах, чтобы удовлетворить все эти нудные запросы ОС, описанные выше.

```
def getattr(self, path):
  st = MyStat()
  st.st_atime = int(time.time())
  st.st_mtime = st.st_atime
  st.st_ctime = st.st_atime
  path_element= path.split('/')
  if path == '/':
    st.st_mode = stat.S_IFDIR | 0755
    st.st_nlink = 2
    st.st_size=4096
  elif path_element[1] == 'odt':
    if len(path_element) == 2 :
      #это каталог
      st.st_mode = stat.S_IFDIR | 0755
      st.st_nlink = 2
    else :
      #это файл
      #он в кэше?
      tpath =os.path.join(cachedir, path[1:])
      if os.path.exists(tpath):
        real=os.stat(tpath)
        st.st_size=real.st_size
      else:
        st.st_size= 2048
      st.st_mode = stat.S_IFREG | 0444
      st.st_nlink = 1
  else:
    return -errno.ENOENT
  return st
```

Последний метод – чтение файла. Если мы передаем **filehandle**, то все просто:

```
def read(self, path, size, offset=0, filehandle=None):
  filehandle.seek(offset)
  buffer= filehandle.read(size)
  return buffer
```

Некоторые методы чтения запрашивают конкретный кусок данных, поэтому мы сдвигаем указатель на смещение **offset** и читаем должный кусок.

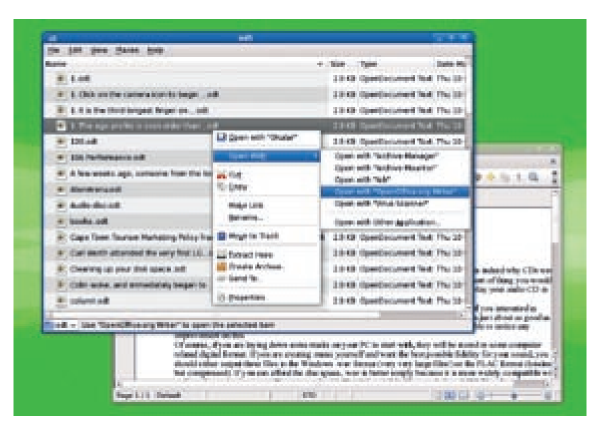
Теперь у нас есть работающая файловая система, которую мы можем примонтировать и читать файлы прямо с Google Docs, но ее легко расширить также и на другие типы файлов. Для выгрузки файлов лучше всего создать новый листинг каталога с правами на запись, а затем нам потребуется использовать методы **mkdir()**, **mknod()** и **write()** для создания записей.

Для монтирования виртуального диска необходимо создать пустой каталог, который будет действовать как точка монтирования, затем вызвать приложение с аргументом **ваша_точка_монтирования**:

```
python simplefuse.py ваша_точка_монтирования
```

Теперь вы можете использовать файловую систему в обычных командах или приложениях. Если вы работаете в современном графическом окружении, ваша смонтированная файловая система должна отображаться в виде устройства, или вы можете просто перейти к ней при помощи менеджера файлов.

Вы можете сделать с ней намного больше – все в web, имеющее API, это отличная игра; а не то создайте более абстрактные файловые системы для конвертирования и реорганизации данных. Просто подумайте, и мы уверены, что у вас появятся идеи множества интересных проектов, для воплощения новых навыков. LXF



› Вы можете загружать документы прямо в просмотрщик или приложение. *OpenOffice.org* легко сделает это для вас.



Скорая помощь

Вы можете отмонтировать файловую систему Fuse командой **fusermount -u ваша_точка_монтирования**, но сперва закройте все открытые файлы.

» Через месяц Попаримся с автоматической генерацией карты температур.

GStreamer: Ваш

Пусть Linux и не испытывает недостатка в таких настольных приложениях, как проигрыватели мультимедиа – **Дмитрий Мусаев** все равно покажет вам, как написать еще один, свой собственный.



Наш эксперт

Дмитрий Мусаев
Программист, проработавший год на видеоканале, который наши московские читатели смотрят по дороге на работу. А вы знали, что он управляется Linux?

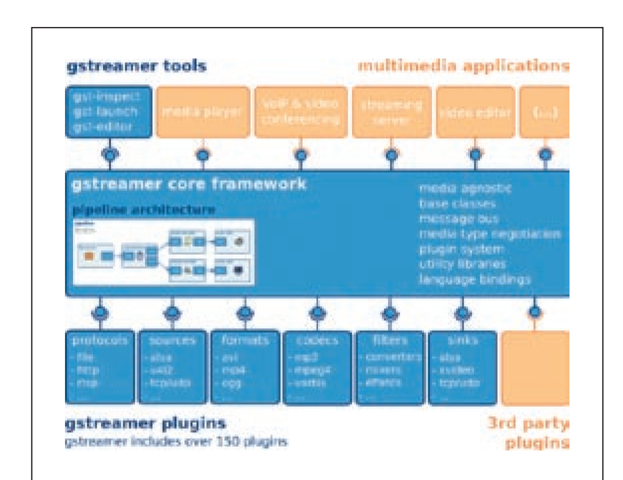
Не возникало ли у вас когда-нибудь желания написать свой собственный медиа-плейер? Дело не только в том, чтобы почить на лаврах *Totem* и *Kaffeine* – это еще и прекрасный повод познакомиться с мультимедиа-каркасом *GStreamer* (gstreamer.freedesktop.org). Он написан на C и имеет интерфейсы для многих других языков программирования, таких как C++, Python и C#. На данный момент для него существует более 150 модулей расширения (plugin), позволяющих декодировать практически все аудио- и видеоформаты (полный список доступен по адресу http://gstreamer.freedesktop.org/documentation/plugins.html). С помощью этих модулей можно не только просматривать или прослушивать аудио- и видеофайлы, но и перекодировать их; например, вы сможете легко написать скрипт для конвертации фильма в формат, понимаемый вашим сотовым телефоном, если последний вообще умеет воспроизводить видео. Можно получать и отправлять медиа-потоки через сеть – в списке модулей вы найдете реализацию нескольких протоколов. Например, чтобы получить видео с web-камеры, достаточно набрать в терминале команду **gst-launch v4l2src! xvimagesink**.

Впечатляет? Тогда давайте перейдем от слов к делу.

Немного теории

Общая архитектура каркаса GStreamer представлена на рис. 1. Как можно видеть, он состоит из базового ядра, утилит и подключаемых модулей. Давайте введем некоторые понятия.

Элемент [element] – наиболее важный компонент в *GStreamer*. Мы будем создавать цепочки связанных между собой элементов и направлять через них поток данных. Элементы соединяются **коннекторами** [pads]; это не вполне точный перевод, но мне кажется, он удобнее, чем «пады» или «подушки». Коннекторы бывают входными [sink pad] и выходными [source pad]. Элемент может иметь различное количество коннекторов: одни присутствуют всегда, другие создаются в зависимости от типа медиа-данных, которые проходят через элемент. Посмотреть, какие именно коннекторы доступны у данного элемента, можно



› Рис. 1. Архитектура каркаса GStreamer: как видно, он имеет три уровня.

при помощи утилиты *gst-inspect*. Выполните команду **gst-inspect decodebin**, и вы получите много интересной информации.

```
Pad Templates:
  SRC template: 'src%d'
    Availability: Sometimes
    Capabilities:
      ANY
  SINK template: 'sink'
    Availability: Always
    Capabilities:
      ANY
```

Мы видим, что у элемента есть постоянный входной коннектор “sink”, совместимый с любым типом медиа-данных, и иногда выходные коннекторы “src%d”, количество которых зависит от типа входных данных. Отметим также, что элемент посылает три сигнала (на них мы остановимся позже):

```
Element Signals:
  "new-decoded-pad" : void user_function (GstElement* object,
                                          GstPad* arg0,
                                          gboolean arg1,
                                          gpointer user_data);
  "removed-decoded-pad" : void user_function (GstElement* object,
                                              GstPad* arg0,
                                              gpointer user_data);
  "unknown-type" : void user_function (GstElement* object,
                                       GstPad* arg0,
                                       GstCaps* arg1,
                                       gpointer user_data);
```

Контейнер [bin] – это объект для управления набором элементов. Контейнер позволяет объединять несколько связанных элементов в один логический. Все, что справедливо для элементов, справедливо и для контейнеров. Например, с помощью контейнеров можно заранее подготовить в программе наборы элементов для кодирования или декодирования различных форматов входных данных и потом, в зависимости от типа последних, использовать тот или иной контейнер.

Конвейер [pipeline] – это специальный подтип контейнера, который позволяет управлять всеми дочерними контейнерами и элементами. Конвейер должен быть контейнером самого верхнего уровня; он присутствует во всех приложения работающих с каркасом.

Элементы *GStreamer* могут находиться в одном из четырех состояний:

» **GST_STATE_NULL** Состояние по умолчанию. В этом состоянии элемент освобождает все ресурсы, которые он занимал.

» **GST_STATE_READY** В этом состоянии элемент размещает глобальные ресурсы. Поток данных закрыт, и позиция в нем выставлена в начало.

» **GST_STATE_PAUSED** В этом состоянии элемент открывает поток, данные подготавливаются для обработки. Элемент позволяет менять позицию в потоке.

видеоплейер



» **GST_STATE_PLAYING** В этом состоянии запускается обработка данных.

GStreamer является многопоточным каркасом, и чтобы упростить взаимодействие приложения и конвейера, в нем введена простая система передачи сообщений **Bus** (шина сообщений). Каждый конвейер автоматически создает шину сообщений – приложению остается только назначить обработчики сигналов и реагировать на интересующие его события. Обработчики могут быть синхронными и асинхронными; мы будем использовать оба типа. Асинхронный обработчик сигналов вызывается в контексте главного цикла *Gtk*-приложения.

GStreamer предоставляет два высокоуровневых контейнера – это Playbin и Decodebin. Они обеспечивают всю рутинную работу по определению типа медиа-данных и их декодированию. Playbin – готовый медиа-плейер, которому нужно указать лишь источник данных и дескриптор окна (что, впрочем, не обязательно – он может создать и свое собственное), куда будет выводиться видео. Но мы будем использовать Decodebin, так как он поддается более тонкой настройке.

Исходя из вышесказанного, давайте составим схему нашего будущего медиа-плейера. Графически она представлена на рис. 2, а в текстовом виде может выглядеть так:

```
gst-launch filesrc location=bubble_dancer.wmv !decodebin
name=decoder decoder. !queue !videoscale !xvimagesink
decoder. ! queue !audioconvert !alsasink
```

Gst-launch – одна из ключевых вспомогательных утилит каркаса. Она позволяет размещать элементы на конвейере, восклицательный знак служит их разделителем: SRCELEMENT.PAD1! SINKELEMENT.PAD1. Если коннекторы не указаны, то перебираются все коннекторы и соединяются подходящие. Свойство **name** используется для задания имени элемента, что дает нам возможность обратиться к нему. При использовании имени элемента точка в конце обязательна – таков синтаксис команды.

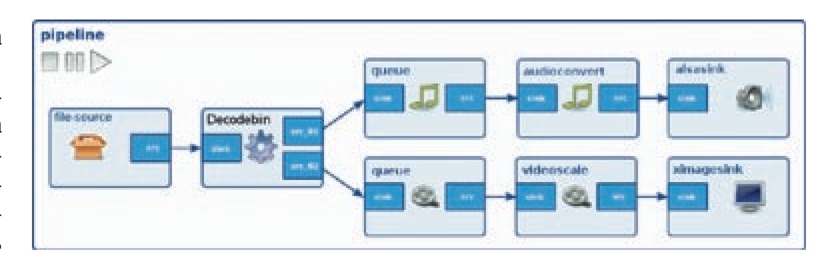
Время кодировать

Перейдем к написанию кода. Я буду использовать C++ и среду разработки *Anjuta*; при желании, вы можете обойтись простым текстовым редактором. Итак, запускаем *Anjuta* и создаем новый проект; на вкладке C++ выбираем GTKmm.

Введите имя проекта, например, Playermm, и выберите его местоположение на жестком диске. Готово: мы имеем функцию **main()** и Glade-форму нашего будущего приложения. Нажимаем F7 и соглашаемся с тем, что нам надо создать проект; после завершения компиляции нажимаем F3. После запуска приложения вы должны увидеть пустое окно с заголовком «Hello world!”.

Теперь внесем небольшие изменения в **main.cc** (выделены жирным шрифтом)

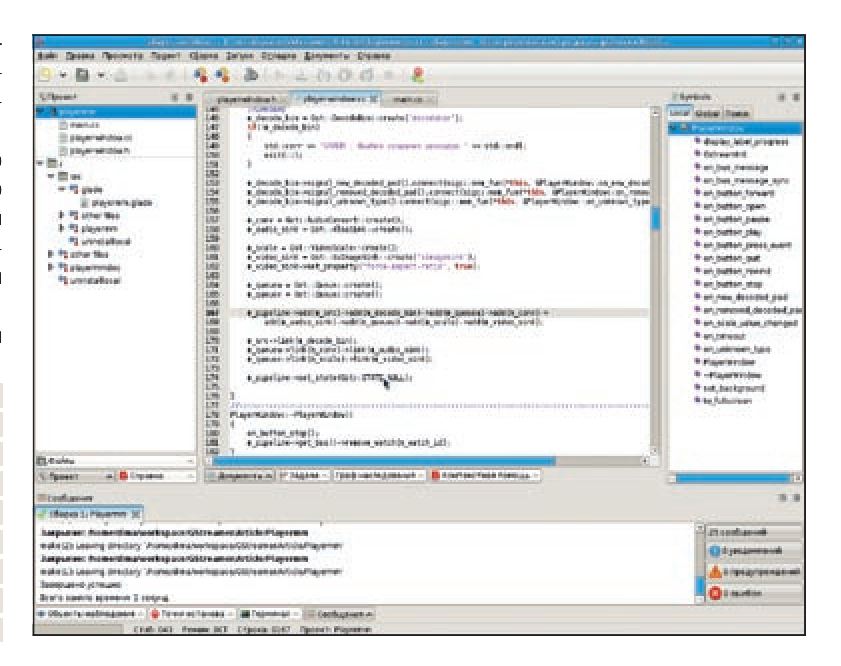
```
int main (int argc, char *argv[]) {
  Gtk::Main kit(argc, argv);
  //Загрузить Glade-файл и создать его виджеты:
  Glib::RefPtr<Gnome::Glade::Xml> refXml;
  try{
    refXml = Gnome::Glade::Xml::create(GLADE_FILE);
  } catch (const Gnome::Glade::XmlError& ex) {
    std::cerr << ex.what() << std::endl;
    return 1;
  }
  PlayerWindow* main_win = 0;
  refXml->get_widget_derived("main_window", main_win);
  if (main_win){
    kit.run(*main_win);
  }
  delete main_win;
  return 0;
}
```



› **Рис. 2. Схема разрабатываемого нами медиа-плейера.**

Откройте файл **playermm.glade** двойным щелчком мыши и разместите на форме вертикальный контейнер, состоящий из трех элементов. В самый верхний добавьте меню, в следующий элемент добавьте еще один вертикальный контейнер, состоящий из четырех элементов, и назовите его 'playerbox'; в самый нижний элемент поместите строку состояния. В первый элемент контейнера playerbox поместите Gtk::DrawingArea – в эту область будет выводиться наше видео. Во второй элемент добавьте метку – Gtk::Label, далее разместите горизонтальную шкалу Gtk::HScale. В самом последнем элементе располагается горизонтальная группа кнопок Gtk::HButtonBox: добавьте туда шесть штук и назовите их btn_play, btn_pause, btn_stop, btn_rewind, btn_forward и btn_open. В результате должен получиться интерфейс, показанный на рис. 4.

Создадим новый класс (Файл > Новый > Класс C++), введем название – PlayerWindow, базовый класс – Gtk::Window. Далее, добавим в заголовочный файл объявления всех виджетов,



› **Рис. 3. Для написания медиа-плейера подойдет Anjuta или любая другая среда, ориентированная на GNOME.**

что мы будем использовать согласно рис. 4 (подробности ищите на прилагаемом DVD).

Теперь, глядя на рис. 2, создадим для каждого его элемента объявление:

```
Glib::RefPtr<Gst::DecodeBin> m_decode_bin;
Glib::RefPtr<Gst::VideoScale> m_scale;
Glib::RefPtr<Gst::XvImageSink> m_video_sink;
Glib::RefPtr<Gst::AudioConvert> m_conv;
Glib::RefPtr<Gst::AlsaSink> m_audio_sink;
Glib::RefPtr<Gst::Pipeline> m_pipeline;
Glib::RefPtr<Gst::FileSrc> m_src;
Glib::RefPtr<Gst::Queue> m_queuev;
Glib::RefPtr<Gst::Queue> m_queuea;
```

Поясним, что означает каждая из этих переменных:

» **FileSrc** Источник данных, то есть файл.

» **DecodeBin** Элемент, который будет раскодировать наши медиаданные. В зависимости от типа данных у него будут создаваться выходы для видео- и аудиопотоков, каждый из которых мы будем передавать в элемент Queue [очередь].

» **Queue** Элемент, который используется в *GStreamer* для создания многопоточности; наши аудио и видео будут обрабатываться в разных потоках.

» **AudioConvert** Конвертирует буфер «сырого» звука ['raw audio'] между различными возможными форматами. Строго говоря, в нашем случае он не нужен и добавлен «для массовости», чтобы вы могли лучше представить себе структуру типового приложения *GStreamer*. Нужно помнить, что разные входные коннекторы могут принимать разные форматы, и даже одни и те же коннекторы могут принимать разные форматы на разных машинах. Поэтому лучше добавлять в цепочки для обработки данных конвертирующие элементы, такие как *audioconvert* и *audioresample* для звука или *ffmpegcolorspace* для видео. Об этом, по крайней мере, предупреждает документация.

» **VideoScale** Изменяет размер изображения. По умолчанию используется билинейный алгоритм, что дает при масштабировании более приятную картинку. В нашем случае элемент также необязательный, поскольку протокол Xv пытается привлечь для масштабирования графическую карту.

» **AlsaSink** В качестве аудиовыхода используется *ALSA* (Advanced Linux Sound Architecture).

» **XvImageSink** Элемент транслирует видеофреймы в нечто, пригодное к выводу на локальный дисплей с использованием видеоконтроллера для преобразования цветов и размера изображения. Также может использоваться для преобразования яркости, контрастности и оттенка цветов. XImageSink для всех этих операций применяет другие элементы, например, VideoScale.

» **Pipeline** Конвейер, в который помещаются все остальные элементы и который осуществляет управление ими.

Кроме того, в нашем заголовочном файле присутствуют конструктор, деструктор и обработчики сигналов.

Вы, наверное, обратили внимание на использование интеллектуального указателя с подсчетом ссылок – **Glib::RefPtr< t_CppObject>**. Это стандартная практика управления ресурсами *glibmm* и *gtkmm*. Суффикс «mm» в конце имени библиотеки, кстати, говорит о том, что это C++-обертка.

› **Рис. 4. Иерархия виджетов и внешний вид окна плейера.**

Свет, камера, мотор!

Заголовочный файл готов; приступим к реализации. Конструктор настраивает все виджеты, связывает виджеты с обработчиками сигналов и выставляет кнопки в начальное состояние, то есть блокирует все, кроме кнопки Открыть. Настройку *GStreamer* вынесем в отдельную функцию, которая будет вызываться из нашего конструктора.

В методе **GstreamInit()** происходит инициализация *GStreamer* и настройка всех нужных нам элементов:

```
void PlayerWindow::GstreamInit() {
 // инициализация GStreamer
 Gst::init();
 //создаем конвейер
 m_pipeline = Gst::Pipeline::create("pipeline");
 //шина сообщений
 Glib::RefPtr<Gst::Bus> bus = m_pipeline->get_bus();
 // Разрешить синхронное извлечение сообщений
 bus->enable_sync_message_emission();
 // Назначить синхронный обработчик сообщений
 bus->signal_sync_message().connect( sigc::mem_fun(*this,
   &PlayerWindow::on_bus_message_sync));
 // Назначить асинхронный обработчик сообщений
 m_watch_id = bus->add_watch(sigc::mem_fun(*this,
   &PlayerWindow::on_bus_message) );
 //источник данных
 m_src = Gst::FileSrc::create("source");
```

Теперь создадим наш декодер. Выше мы рассматривали вывод команды **gst-inspect decodebin** и видели, что он посылает три сигнала:

» **new-decoded-pad** Создан новый коннектор. В этом обработчике мы будем связывать аудио- и видеоцепочки с выходными данными декодера, исходя из типа создаваемого коннектора (см. рис. 2).

» **removed-decoded-pad** Коннектор удален.

» **unknown-type** Неизвестный тип медиа-данных.

Из всех этих событий нам нужно обрабатывать только создание новых коннекторов.

```
 m_decode_bin = Gst::DecodeBin::create("decodebin");
 m_decode_bin->signal_new_decoded_pad().
connect(sigc::mem_fun(*this,
   &PlayerWindow::on_new_decoded_pad));
```

Следующим шагом создаются оставшиеся элементы:

```
 m_conv = Gst::AudioConvert::create();
 m_audio_sink = Gst::AlsaSink::create();
 m_scale = Gst::VideoScale::create();
 m_video_sink = Gst::XvImageSink::create("ximagesink");
 m_video_sink->set_property("force-aspect-ratio", true);
 m_queuea = Gst::Queue::create();
 m_queuev = Gst::Queue::create();
```

Наконец, мы размещаем все элементы на конвейере и связываем их между собой:

```
 m_pipeline->add(m_src)->add(m_decode_bin)->add(m_queuea)->add(m_conv)->
   add(m_audio_sink)->add(m_queuev)->add(m_scale)->add(m_video_sink);
 m_src->link(m_decode_bin);
 // аудиоветвь
 m_queuea->link(m_conv)->link(m_audio_sink);
 // видеоветвь
 m_queuev->link(m_scale)->link(m_video_sink);
 m_pipeline->set_state(Gst::STATE_NULL);
}
```

Далее, реализуем обработчик сигналов для декодера. В случае, если речь идет о создании нового коннектора, его текст представлен ниже. В имени вновь созданного коннектора мы ищем строку «video» или «audio», далее получаем входной коннектор соответствующей ветви элементов и связываем их. Наконец, мы проверяем результат связывания, и в случае неудачи сообщаем об этом на консоль программы.

```
void PlayerWindow::on_new_decoded_pad(const
Glib::RefPtr<Gst::Pad>& pad,
                                    bool arg1) {
 if (pad->get_caps()->get_structure(0).get_name().find("video")
    != Glib::ustring::npos) {
  Glib::RefPtr<Gst::Pad> sinkPad = m_queuev-
>get_static_pad("sink");
  // связать только один раз
  if (!sinkPad->is_linked()) {
   Gst::PadLinkReturn ret = pad->link(sinkPad);
   if (ret != Gst::PAD_LINK_OK
     && ret != Gst::PAD_LINK_WAS_LINKED) {
    std::cerr << "Невозможно назначить видеовыход" <<
std::endl;
   }
  }
 }
 //... подключение аудиопотока осуществляется точно так же
}
```

Каждый обработчик сигналов принимает одинаковые последовательности сообщений: первым их получает синхронный обработчик, далее – асинхронный. Рассмотрим синхронный обработчик шины сообщений:

```
void PlayerWindow::on_bus_message_sync( const
Glib::RefPtr<Gst::Message>& message) {
 // игнорировать все события кроме 'prepare-xwindow-id'
 if(message->get_message_type() != Gst::MESSAGE_ELEMENT
    && !message->get_structure().has_name("prepare-xwindow-
id"))
  return;
 Glib::RefPtr<Gst::Element> element =
  Glib::RefPtr<Gst::Element>::cast_dynamic(message-
>get_source());
 Glib::RefPtr< Gst::ElementInterfaced<Gst::XOverlay> > xoverlay =
  Gst::Interface::cast <Gst::XOverlay>(element);
 if(xoverlay){
  const gulong xWindowId =
   GDK_WINDOW_XID(m_video_area->get_window()->gobj());
  xoverlay->set_xwindow_id(xWindowId);
 }
}
```

Задача этого обработчика – получить контекст окна, в которое будет выводится видео. Наш асинхронный обработчик, **on_bus_message()**, обрабатывает только два сигнала: это конец потока данных и ошибка. В любом случае обработчик вызывает метод **on_button_stop()**, который переводит конвейер в состояние STATE_NULL.

Вот почти и все: осталось только добавить обработчики нажатия кнопок, и приложение можно запускать. Все управление воспроизведением сводится к изменению состояния контейнера посредством шести кнопок. Также в обработчиках кнопок разместим управление таймером. Он нужен нам для обновления прогресса воспроизведения и отслеживания прошедшего времени. Например, обработчик кнопки «Play» может выглядеть так:

```
void PlayerWindow::on_button_play() {
 //изменить состояние кнопок
 m_progress_scale->set_sensitive();
 m_play_button->set_sensitive(false);
 m_pause_button->set_sensitive();
 m_stop_button->set_sensitive();
 m_rewind_button->set_sensitive();
 m_forward_button->set_sensitive();
 m_open_button->set_sensitive(false);
 m_play_button->hide();
 m_pause_button->show();
 // вызывать функцию on_timeout каждые 200 мс
 // для регулярного обновления позиции в потоке
 m_timeout_connection = Glib::signal_timeout().connect(
  sigc::mem_fun(*this, &PlayerWindow::on_timeout), 200);
 // Включить режим воспроизведения
 m_pipeline->set_state(Gst::STATE_PLAYING);
}
```

Соответственно, постановка на паузу будет выглядеть так:

```
void PlayerWindow::on_button_pause() {
 m_play_button->set_sensitive();
 m_pause_button->set_sensitive(false);
 m_pause_button->hide();
 m_play_button->show();
 // Остановить таймер
 m_timeout_connection.disconnect();
 // Пауза
 m_pipeline->set_state(Gst::STATE_PAUSED);
}
```

Полный текст приложения имеется на прилагающемся к журналу диске. Кроме того, пакеты *GStreamer* и *GStreamermm* содержат примеры, которые помогают понять все тонкости использования данного каркаса. Проект активно развивается, расширяется документация (http://gstreamer.freedesktop.org/documentation/), в планах есть интеграция с KDE, что упростит обработку событий; об этом вы можете подробнее прочитать на сайте проекта.

Ну и, прежде чем закончить статью, давайте посмотрим, что у нас получилось! Для запуска приложения скопируйте его в свою рабочую папку, запустите *Anjuta*, откройте диалог Настроить проект (Сборка > Конфигурация проекта...) отметьте галочку Пересоздать проект и нажмите Выполнить. Теперь *Playermm* можно запустить, нажав F3. LXF



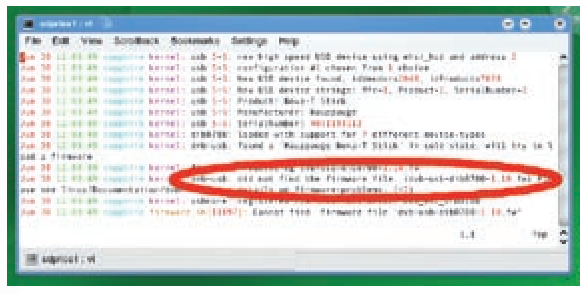
› Рис. 5. Результат наших трудов: не слишком развитый, но вполне работоспособный видеоплейер.

» Пропустили номер? Узнайте на с. 107, как получить его прямо сейчас.

Perl: Запишем

Автоматизируйте запись вашего ТВ с помощью Perl. **Стив Прайс** покажет, как просто создать мультитюнеровый ТВ-рекодер.





Без паники! Просто поищите необходимый ядру файл прошивки и скопируйте его в правильный каталог.



Наш эксперт

Стив Прайс пишет Perl-скрипты уже много лет и является разработчиком CPAN. Сейчас он знает о цифровых стандартах вещания больше, чем даже хотел – исключительно потому, что всегда боялся пропустить очередную серию «Теории Большого взрыва».

Вы наверняка когда-нибудь да хотели сделать автоматическую запись телепередач на компьютере с Linux – дело стало только за правильной установкой *MythTV*. А как насчет свободной программы, которая автоматически составит расписание записей ваших любимых передач? Или – как вам идея получить готовый цифровой видеомагнитофон, всего-навсего воткнув в USB-разъем новое устройство? Если на оба эти вопроса вы ответили «Да», то этот урок – для вас: здесь я покажу, как использовать Perl'овский модуль (**Linux::DVB::DVBT**) для управления DVB-t ТВ-тюнерами.

Прежде всего решите, какой тюнер купить. Я рекомендую USB-устройство Hauppauge WinTV-NOVA-t (его обзор см. в **LXF106**), простое в настройке и обеспечивающее хорошее качество картинки. Дополнительным преимуществом использования USB-устройств по сравнению со встраиваемыми картами является возможность добавлять тюнеры в любой момент, и это не повлияет на компьютер. Перед походом в магазин, однако, не мешает выяснить, какие тюнеры поддерживаются вашим дистрибутивом. Запустив приложение по настройке ТВ-карты, можно получить много полезной информации: например, выведется список поддерживаемых производителей и моделей, которые должны работать на вашей машине.

Правильное оборудование

Какой бы тюнер вы ни купили, я не устану подчеркивать, насколько важно использовать хорошую антенну, а не ту, что поставляется с тюнером. У вас будет намного лучшая картинка (да и с начальной настройкой будет полегче), если антенна дает хороший сигнал. Простейший способ отделаться – подключить к имеющейся антенне усилитель с несколькими выходами, а на один из свободных портов навесить новый тюнер.

Часть 1: Установка

Обзаведясь тюнером и вставив его в свою систему, запустите утилиту установки ТВ-карты и выберите свой тюнер в списке. Этого должно хватить. Заметьте, я сказал «должно»: если возникнет проблема, то, скорее всего, у вас не загружена прошивка карты, и установить ее придется самому. Не пугайтесь: это проще, чем кажется. Сначала взгляните на сообщения в **/var/log/messages** и поищите там указания на доступ к аппаратной части тюнера. Если вы нашли сообщение типа «Did not find the firmware file», то вам придется поискать в сети двоичный файл прошивки, на который указывает то сообщение. Скорее всего, это будет файл с расширением FW (например, моему тюнеру Hauppauge понадобился файл прошивки под названием **dvb-usb-dib0700-1.10.fw**). Просто погуглите данное имя файла; сайт производителя или вики *MythTV* – также отличные места для поиска.

Отыскав и скачав файл прошивки, скопируйте его в каталог прошивок вашего дистрибутива (обычно это **/lib/firmware**). Например:

```
sudo cp dvb-usb-dib0700-1.10.fw /lib/firmware
```

Месяц назад Звучала музыка из потока мультимедиа, созданного UPnP.

любую передачу

После установки прошивки нужно, чтобы ядро загрузило ее в тюнер. В случае USB-устройства просто отсоедините его и присоедините обратно; в случае карт PCI вам придется перезагрузиться. Если карта определится правильно, то в файле сообщений вы увидите фразу типа «Firmware started successfully». Можете побаловать себя чашечкой кофе, прежде чем перейти ко второму этапу, но пока чайник еще не вскипел, давайте сделаем отступление и пообсуждаем поддержку устройств.

Во время работы ваш тюнер фигурирует как отдельное устройство в каталоге **/dev**, почти как CD-ROM. Однако, в отличие от CD-ROM'а, тюнер состоит из нескольких устройств, каждое из которых отвечает за свой аспект приема ТВ-сигнала (я буду называть их набором устройств). Эти устройства объединены в каталог под названием **adapter**, который представляет тюнер как целое. Так как к системе может быть подключено несколько тюнеров, каждый из них представлен своим каталогом адаптера, которые нумеруются, начиная с 0. Заглянув в каталог устройств, вы увидите, что его структура имеет примерно такой вид:

```
/dev/dvb/
adapter0/
demux0 dvr0 frontend0
adapter1/
demux0 dvr0 frontend0
demux1 dvr1 frontend1
```

Устройства из каталога **adapter** контролируют разные аспекты тюнера: **frontend** выбирает правильную полосу частот; **demux** управляет выбором канала; а **dvr** выдает сырой поток вещания (видео), используемый при записи передач. Как и при нумерации адаптеров, если в адаптере имеется несколько приемников, то существует несколько наборов устройств, и они нумеруются, начиная с 0.

Модуль Perl обеспечивает интерфейс для этих устройств. Если таковых несколько, нужно указать номер адаптера и номер устройства, которое вы хотите использовать. Для указания номера набора устройств модуль Perl использует поле **frontend_num** (подробности ниже).

Установка

Теперь, когда ваше оборудование заработало, перейдем к установке ПО. Самый легкий способ сделать это – зайти от имени root и запустить скрипт *cpan,* который поставляется с установкой Perl'а:

```
sudo cpan Linux::DVB::DVBT
```

Будет установлен модуль **Linux::DVB::DVBT**, а заодно автоматически учтутся все требуемые зависимости. Обратите внимание, что при первом запуске установки со CPAN вам зададут много вопросов. Ответов, предлагаемых по умолчанию, обычно достаточно для правильных настроек.

Альтернатива – скачать модуль из CPAN и выполнить ручную компиляцию и установку (правда, отсутствующие зависимости придется накопать самим):

```
perl Makefile.PL
make test
sudo make install
```



› Мультитюнеровая запись при помощи USB-устройств. Бог ниспослал нам лишние USB-разъемы именно для них.

При этом не только добавится **Linux::DVB::DVBT**, но вас еще и наградят несколькими скриптами-примерами (*dvbt-scan*, *dvbt-record* и *dvbtepg*). Эти сценарии – полностью работающие варианты тех функций, про которые я расскажу на данном уроке, и если вы не хотите пачкаться с написанием скриптов, можете использовать их.

Установив модуль, можно протестировать его, получив информацию о ТВ-тюнере. **Linux::DVB::DVBT** обеспечивает объектно-ориентированный интерфейс: как правило, вы создаете экземпляр класса, а затем используете методы данного экземпляра. Для разнообразия, метод **device_list()**, который мы собираемся рассматривать, использует в качестве экземпляра модуль сам по себе (если вы не знакомы с объектно-ориентированным программированием, бояться нечего: куски кода в примерах прояснят всю картину).

Метод **device_list()** возвращает массив хэшей; каждый хэш содержит сведения об установленных тюнерах. Эту информацию можно отобразить посредством следующего кода:

```
#!/usr/bin/perl -w
use Linux::DVB::DVBT;
# получить список установленных адаптеров
my @devices = Linux::DVB::DVBT->device_list() ;
foreach (@devices)
{
 printf "%s : adapter number: %d, frontend number: %d\n",
  $_->{name}, $_->{adapter_num}, $_->{frontend_num} ;
}
```

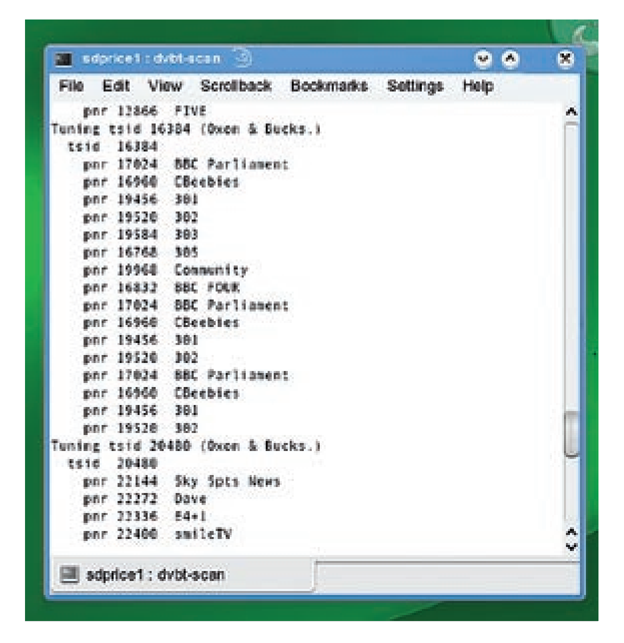
В результате в виде списка отобразятся номера **adapter** и **frontend** для каждого тюнера, а также название изготовителя. Данные номера понадобятся вам, если вы хотите использовать мультитюнеровую запись, так что приберегите их до поры, когда мы этим займемся.

Если ваш тюнер работает, однако никакого вывода о нем не видно, значит, что-то не так. Вернитесь назад и просмотрите ваши файлы журналов на предмет сообщений об ошибках (может быть, вы забыли, что ядро должно перезагрузить прошивку тю- »

Скорая помощь

Полезно помнить: каждый час записи требует примерно 1 ГБ дискового пространства. Так что озаботьтесь покупкой большого диска исключительно под эти нужды (вы ведь не хотите, чтоб ваш сервер рухнул под тяжестью записи Дома-2, ха-ха).

› Нет вести «Получилось!» доходчивее, чем список частот вещания.



нера). Иначе – примите сердечные поздравления: среда для записи передач с вашего ТВ-тюнера готова к работе, и пришло время перейти к чему-нибудь более интересному. Так как мы уже установили аппаратную часть и ПО, и почти закончили с настройкой, то осталась одна задача: настройка оборудования на вещаемые каналы.

Настройка

Самая тяжелая часть данного этапа – раздобыть файл частот, содержащий информацию о вашей местной сети. Тут есть две возможности: если у вас установлен пакет **dvb**, вы найдете файлы в **/usr/share/dvb/dvb-t**; если у вас установлен *Kaffeine*, они будут в **$HOME/.kde/share/apps/kaffeine/dvb-t**; а не то возьмите их с DVD этого месяца.

Отыскав каталог, выберите файл, соответствующий локальному вещанию. Простейший способ [работает только для Великобритании, – *прим. пер.*] – зайти на www.ukfree.tv/transmitters.php и ввести свой почтовый адрес. Например, для моего местного вещателя это Оксфорд, так что я использую файл **uk-Oxford**. Вы сразу поймете, что у вас правильный файл, когда увидите в окне терминала список названий каналов. Передайте путь вашего файла частот методу **scan_from_file()**, откиньтесь на спинку стула и подождите завершения. Соответствующий скрипт выглядит так:

```
use Linux::DVB::DVBT ;
## Создаем dvb-объект (используется первый найденный адаптер).
my $dvb = Linux::DVB::DVBT->new() ;
## Отображать прогресс сканирования
Linux::DVB::DVBT->verbose(1) ;
## Выполнить сканирование (путь подставьте свой)
$dvb->scan_from_file('/usr/share/dvb/dvb-t/uk-Oxford') ;
```

Вместо этого можно использовать установленный скрипт

```
# Опять подставьте свой путь
dvbt-scan /usr/share/dvb/dvb-t/uk-Oxford
```

Обратите внимание, что у всех скриптов одна и та же структура: импорт модуля **Linux::DVB::DVBT** при помощи ключевого слова **use**; создание нового объекта DVB методом **new()**; использование методов объекта DVB для некоторых действий.

Метод **scan_from_file()** использует файл частот, чтобы сканировать информацию о вещании: он собирает детали о вещании всех каналов этого передатчика и автоматически записывает ее в файлы настроек в **$HOME/.tv**. Последние будут использоваться при всех последующих вызовах, так что достаточно запустить его только один раз (даже если у вас подключено несколько тюнеров).

Часть 2: Запись

Теперь, проделав всю работу по настройке, можно приступить к тому, ради чего мы здесь собрались: записи передач. Модуль Perl обеспечивает два метода, используемые при записи: метод выбора канала **select_channel()** и метод записи файла **record()**. Для выбора канала надо всего лишь написать его название:

```
$dvb->select_channel('BBC ONE') ;
```

(Приятная особенность **select_channel()** – то, что он допускает вольное написание имени канала. Вы можете написать название с любым числом пробелов, в произвольном регистре, с цифрами вместо числительных.)

Затем вызовем метод **record()**: его аргументы – имя видеофайла и время записи.

```
$dvb->record('./test.ts', '00:30') ;
```

Записываемый файл обычно снабжают расширением **.ts**, но это не обязательно, т.к. видеоинформация хранится в формате MPEG 2 Transport Stream. Длительность записи можно выражать количеством секунд (одно число), или в формате ЧАСЫ:МИНУТЫ:СЕКУНДЫ, или, как в примере, в формате ЧАСЫ:МИНУТЫ.

Вместо того, чтобы писать собственный скрипт, можно использовать скрипт *dvbt-record*, снабдив его теми же параметрами и в том же формате, как описано выше:

```
dvbt-record 'BBC ONE' ./test.ts 00:30
```

Подождите 30 минут, и у вас появится первая записанная телепередача. Естественно, это можно делать и вручную, всякий раз, когда вам захочется записать передачу. Однако гораздо удобнее взвалить эту работу на машину: тут, естественно, пригодится *Cron*.

Отредактируйте свой файл *Cron* с помощью

```
crontab -e
```

Формат файла *Cron* выглядит примерно так:

```
МИНУТА ЧАС ДЕНЬ_МЕСЯЦА МЕСЯЦ ДЕНЬ_НЕДЕЛИ ДЕЙСТВИЕ
```

Обычно для записи время начала передачи (или чуть раньше) задается в формате **МИНУТА/ЧАС**, дата – **ДЕНЬ_МЕСЯЦА/МЕСЯЦ**, а в **ДЕНЬ_НЕДЕЛИ** ставится *. В качестве **ДЕЙСТВИЯ** укажите вызов записывающего Perl-скрипта (или имеющегося у вас скрипта *dvbt-record*). Задав **ДЕНЬ_НЕДЕЛИ**, мы тем самым определим задачу *Cron*, которая будет выполнять запись раз в неделю. Одиночная запись 2 июля 30-минутной передачи на BBC1, начинающейся в 9:00, выглядит как

```
00 21 02 07 * dvbt-record bbc1 ~/news.ts 00:30
```

Одиночная запись часовой передачи, идущей 3 июля на ITV в восемь вечера, но с минутными интервалами в начале и конце, выглядит как

```
58 19 03 07 * dvbt-record itv1 ~/documentary.ts 01:04
```

А вот еженедельная получасовая передача на Пятом канале, начинающаяся в 10 вечера по четвергам (здесь 0=воскресенье, 1=понедельник):

Скорая помощь

Обратитесь к старым журналам Linux Format, и вы многое узнаете о настройке медиа-серверов. Стоит ознакомиться с Mediatomb, Fuppes, Ushare или Twonkyvision.

» **Пропустили номер?** Узнайте на с. 107, как получить его прямо сейчас.

```
00 22 * * 4 dvbt-record five ~/big_bang.ts 00:30
```

Как вы помните, номер **adapter** указывает на карту либо USB-устройство. Номер **frontend** указывает на приемник в адаптере (обычно это 0).

Покамест мы позволяли Perl'у автоматически устанавливать эти номера, в предположении, что тюнер у нас всего один. Если вы махнули рукой на предостережения и купили другой тюнер, то вам, наверно, интересно узнать, как велеть Perl'у использовать это новое устройство. Все, что вам нужно сделать – это указать номера **adapter** и **frontend** при создании нового объекта DVB. Например, для создания объекта DVB, который управляет вторым из ваших тюнеров, пропишите

```
## Создаем dvb-объект (используется адаптер 1, а не 0.
my $dvb2 = Linux::DVB::DVBT->new(
'adapter_num' => 1,
'frontend_num' => 0,
) ;
```

а затем вызывайте методы этого нового объекта, как делалось ранее (заменив **$dvb** на **$dvb2**). Очень просто.

Воспроизведение

Мы задали несколько подходящих задач *Cron,* и вот уже наш жесткий диск ломится от файлов TS. Что делать теперь? Простейший ответ – хотите, смотрите их прямо так: программы воспроизведения видео (наподобие *MPlayer*) прекрасно умеют проигрывать файлы потока MPEG 2. Ну, а если замахнуться на большее? Одна из возможностей – потоковая передача наших записей по дому; в этом случае нужно позаботиться, чтобы формат файлов годился для потоковой передачи медиа-сервером. Некоторые серверы (и клиенты) воротят нос от TS-файлов, и перед воспроизведением понадобится их конвертировать.

Для тех файлов, которые вы хотите раз посмотреть, а затем удалить (или если вам не жалко места на жестком диске), самая быстрая конвертация получится при использовании *FFmpeg*, если перепаковать файл как стандартный файл MPEG 2 командой

```
ffmpeg -i input.ts -async 1 -vcodec copy -acodec copy -y output.mpeg
```

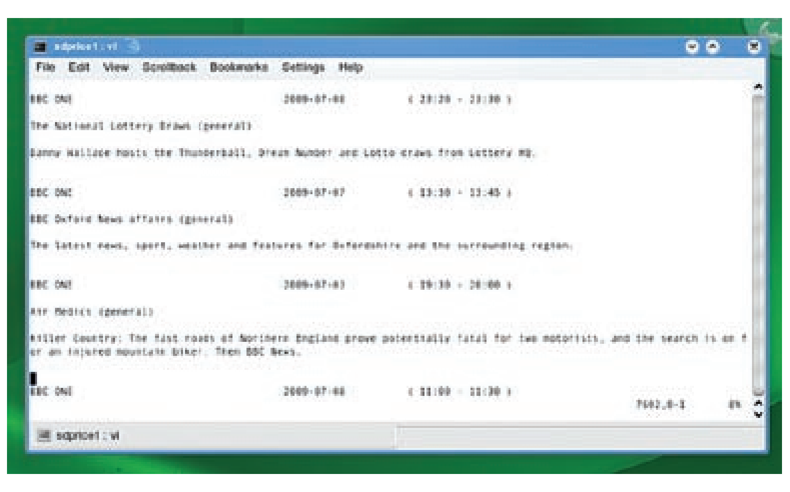
Однако любое видео вы, скорее всего, пожелаете сжать как можно сильнее, сохранив качество картинки. Соответствующие командные строки *FFmpeg*'а имеются на **LXF**DVD, однако вы обнаружите, что процесс этот не быстр – сжатие часового видео может слопать часов шесть!

Программа передач

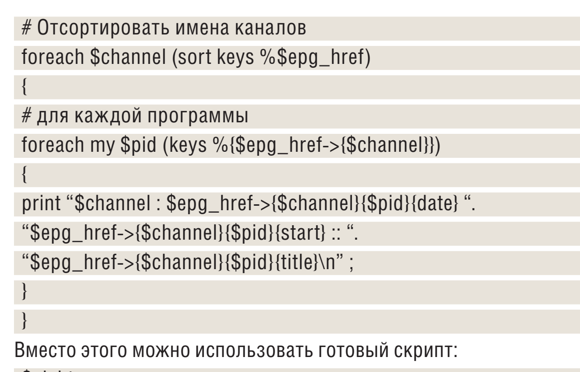
Я закончу этот урок, затронув другую возможность, предоставляемую модулем Perl: электронную программу передач. Вызвав метод **epg()**, можно получить программу передач на следующие две недели. На этих страницах не хватит места, чтобы выложить вам полнофункциональный скрипт для программы передач, но, по крайней мере, основы описать получится.

Вызов метода **epg()** возвращает две порции информации, каждая по своей ссылке на хэш: обзор передач и даты передач. Даты мы опустим (они пригодятся, только если вы держите программу передач в базе данных), а сосредоточимся на информации о передачах (можете обратиться к документации **Linux::DVB::DVBT** за деталями о полной структуре хэшей передач и дат). Каждая передача хранится в иерархии хэшей, проиндексированной по имени канала и по уникальной передаче. Чтобы обработать всю программу передач, мы просто пройдем в цикле все каналы и передачи на каждом канале. Например:

```
## Получить программу передач
my ($epg_href, $dates_href) = $dvb->epg() ;
```



› **Ценный подарок: электронная программа передач.**



```
# Отсортировать имена каналов
foreach $channel (sort keys %$epg_href)
{
# для каждой программы
foreach my $pid (keys %{$epg_href->{$channel}})
{
print "$channel : $epg_href->{$channel}{$pid}{date} ".
"$epg_href->{$channel}{$pid}{start} :: ".
"$epg_href->{$channel}{$pid}{title}\n" ;
}
}
```

Вместо этого можно использовать готовый скрипт:

```
$ dvbt-epg
```

На своей системе я храню EPG-данные в базе данных *MySQL*, а отображаю программу передач в web-приложении, где могу искать передачи и делать расписание записей. Я понял, что мне нужно регулярно обновлять EPG-данные, так как вещатели часто меняют порядок передач по своим капризам. Ежеутреннего обновления данных, скорее всего, достаточно, чтобы время записи было актуальным.

Другая идея, которая вам может понравиться, состоит в создании скрипта, который читает из файла список названий передач. Этот скрипт затем использует последнюю EPG-информацию, чтобы автоматически создавать расписание записи тех передач, названия которых встречаются в файле.

Как видите, запись передач можно сделать простой командной строкой. Также можно заметить, что довольно легко расширить вашу систему до мультитюнерового рекордера, если добавить дополнительные USB-тюнеры. Объедините все вместе, и у вас будет многоканальный рекордер, стоимость которого на порядок меньше стоимости коммерческих решений; а его возможности ограничены только вашим воображением (ну ладно, еще и вашими способностями кодировать в Perl!). **LXF**

Скорая помощь

Для чтения файла сообщений используйте *vi*: он подсвечивает сообщение, и читать его проще.

Уникальные имена файлов

Для повторяющихся выпусков программ легко можно случайно перезаписать старые записи. Во избежание этого, используйте команду *date* для указания имени файла в задаче *Cron*'а, и имена файлов не будут повторяться. Самый простой способ – использовать команду *date* с опцией форматирования, добавляющей год, месяц, день, час и минуты как 12-значное число. Команда выглядит так:

```
dvbt-record five ~/big_bang-`date +"%Y%m%d%H%M"`.ts 00:30
```

» **Через месяц** Фильтруйте сомнительный контент Сети с помощью Privoxy.



Есть вопрос по Open Source? Пишите нам по адресу answers@linuxformat.ru

В этом месяце мы ответим на вопросы про:

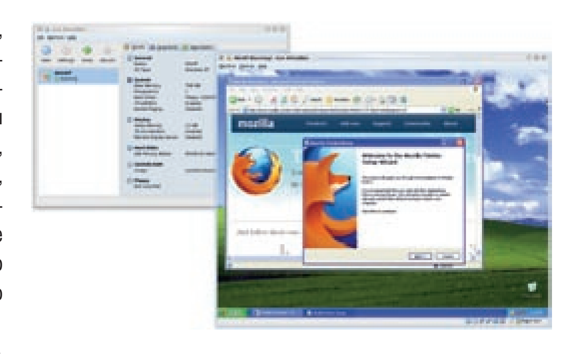
1 Работа изнутри

В Я поставил *VirtualBox* в Linux Mint и установил в нем Windows XP, чтобы работать с неким приложением. Согласно Sun Microsystems, это возможно, на то и *VirtualBox*, но вот как это сделать, они, к сожалению, не сказали. Не могли бы ли вы мне помочь? Должен предупредить, что я полный нуль в Linux, хоть и пользуюсь компьютером много лет.

Деннис Янг [Dennis Young]

О Виртуальная машина, будь то *VirtualBox* или *VMware*, может рассматриваться как вполне самостоятельный отдельный компьютер. Почти полное отделение гостевой ОС (той, что работает в виртуальной машине) от основной хост-системы (той, в которой запущена *VirtualBox*) – это одно из ключевых преимуществ такой настройки. Поэтому все, что Вы хотели бы сделать в гостевой Windows, делайте так, как если бы это было в обычной Windows, установленной на отдельном компьютере. Для установки приложения идите на сайт программы, используя браузер в Windows, скачивайте **setup.exe** (или что там) и запускайте инсталлятор обычным порядком. Работая в виртуальной машине, можно забыть даже о самом существовании хост-системы.

Это разделение чрезвычайно удобно, но также и затрудняет обмен файлами между хостом и гостевой системой. Если, скажем, Вы скачали файл Windows-программы из Linux и Вам нужно перебросить ее в виртуальную машину Windows, то один из способов этого достичь – настроить общую папку в Windows и работать с ней из Linux (гостевая и хост-системы рассматриваются как два разных компьютера в одной сети). Вы не указали, какой дистрибутив и какую графическую среду Вы используете, но в большинстве случаев существует сетевой браузер, позволяющий найти и смонтировать разделяемые папки Windows. Для постоянного использования это будет наиболее удобным способом, но для одноразового доступа проще скопировать файлы на USB-носитель, отмонтировать его в Linux и получить к нему доступ в виртуальной машине в *VirtualBox*. Можно также воспользоваться функцией *VirtualBox* под названием Разделяемые папки [Shared folders], но она потребует установки специальных гостевых дополнений в Вашу виртуальную Windows. **ПХ**



› Виртуальная машина — это практически отдельный компьютер, так что смело устанавливайте программы обычным способом.

Наши эксперты

» Мы найдем ответы на любой вопрос — от проблем с установкой системы или модемом до сетевого администрирования, главное — спросить!



Нейл Ботвик

Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс

Майк был одним из создателей прототипа **LXF** – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.



Пол Хадсон

Пол – местный супер-программист, и он может и хочет управиться со всеми вашими проблемами по части web и баз данных.



Грэм Моррисон

Когда он не обозревает кучи програмного обеспечения и не халтурит с MythTV, Грэм готов дать ответ касательно любого оборудования и проблем виртуализации.



Валентин Синицын

В редкие свободные минуты главный редактор нашего журнала обычно запускает *mcedit*, чтобы отшлифовать какое-нибудь открытое приложение. Его любимая тема – настольный Linux.



Юлия Дронова

Если компьютер у Юлии не занят выполнением команды *emerge*, она спешит применить его для модерирования ЛинуксФорума.

КУДА ПОСЫЛАТЬ ВОПРОСЫ

Пишите нам по адресу: answers@linuxformat.ru **или спрашивайте на форуме:** www.linuxformat.ru

2 Linux особого назначения

В Я работаю в сфере образования и познакомился с Linux примерно год назад, имея дело с нетбуками. До этого я уже некоторое время писал программы управления и собирал интерфейсные устройства (в основном на базе стандарта RS-232), используя старые ноутбуки с картой CompactFlash. Графического интерфейса тут не надо, и я работаю с MS-DOS и C++, а autoexec.bat вполне справляется со всем необходимым. Я жажду переехать на Linux из DOS, но пока не получил ответа на простой вопрос: какова минимальная установка, которая даст мне рабочую командную строку и обеспечит, чтобы моя программа запускалась при загрузке автоматически?

Также, не возникнет ли проблем при обращении моих программ к параллельным и последовательным портам в Linux? Заодно отмечу, что я пробовал Tiny Core Linux и ищу решения аналогичной проблемы. Как заставить браузер автоматически запускаться при загрузке, в рамках создания простого web-киоска для школьного набора ПО?

Гленн Уолш [Glenn Walsh]

О Неплохой выбор минимальной системы – Debian. CD для сетевой установки ставит только базовую систему, а далее по сети устанавливается то немногое, что необходимо именно вам. Существуют и другие «минимальные»

дистрибутивы, достойные рассмотрения – например, Puppy Linux (www.puppylinux.org), но преимущество использования именно Debian состоит в том, что в нем можно настроить и полнофункциональный рабочий стол и развернуть тот же дистрибутив на настольной машине для разработки Ваших программ, без проблем переноса их на целевую систему для тестирования. В сетевой установке Debian используйте базовую текстовую установку и снимите отметку со всех групп пакетов, когда появится запрос о программах, которые Вы хотите установить. Тогда Вы получите только базовую систему.

Что касается языков программирования, то довольно просто разобраться с Python, и у него есть модули для доступа к последовательным и параллельным портам. Раздобыть их можно по адресу http://pyserial.sourceforge.net, и там же имеются пакеты для многих дистрибутивов. Если Вы предпочитаете компилируемый язык, то в Mono есть набор библиотек для последовательных портов (хотя доступ к параллельному порту поддерживается хуже), или же оставайтесь на знакомом Вам C++. Установка программ из командной строки в Debian делается с помощью **apt-get**; добавить Python вместе с модулями для последовательного и параллельного портов можно так:

```
apt-get install python-serial python-parallel
```

Упоминать сам Python не обязательно: он автоматически установится как зависимость для этих двух пакетов.

Запуск программ при старте компьютера осуществляется скриптами из каталога **/etc/init.d** – они выполняют различные службы, необходимые для функционирования ОС. Одна из служб, в свою очередь, запускает локальные скрипты из файла **/etc/rc.local**, так что программы, необходимые Вам при запуске, нужно вписать в этот файл. Каждая из команд должна запускаться в фоновом режиме, поэтому добавляйте знак **&** в конце каждой строки, иначе процесс загрузки не станет продолжаться до тех пор, пока эта команда не отработает. Например:

```
/usr/local/bin/мой_скрипт.py &
```

Автоматический запуск настольного приложения несколько отличается. Большинство оконных менеджеров используют файл **.xinitrc** из домашнего каталога пользователя, так что добавлять команды нужно в него. Более крупные рабочие среды знают про каталог **Autostart**, но для простого киоск-приложения это будет перебор. В Tiny Core все запускается из **.xsession**, поэтому просто допишите команду, открывающую нужный Вам браузер, в конец этого файла.

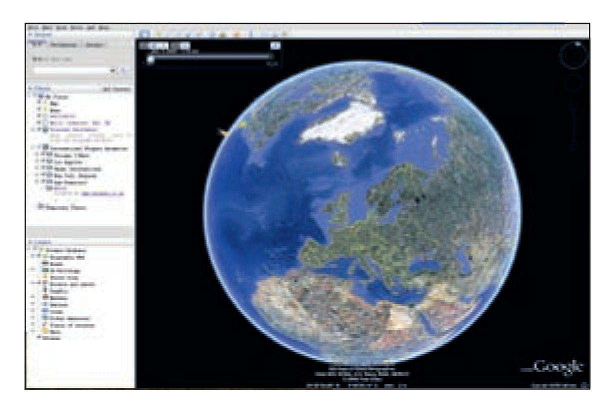
В другом дистрибутиве настроить браузер, предназначенный для работы с киоском, можно, поместив в **/etc/rc.local** (или аналог этого файла в установленном у Вас дистрибутиве – точный путь может несколько отличаться) следующую команду:

```
/bin/su - kioskuser -c "/usr/bin/startx &>/dev/null" &
```

Она запускает графический сервер от имени пользователя **kioskuser** (**rc.local** выполняется от имени суперпользователя-root, так что вводить пароль не требуется). Поскольку браузер будет запускаться в полноэкранном режиме, то менеджер окон Вам и не нужен; для запуска *Firefox* поместите следующие строки в файл **/home/kioskuser/.xinitrc**:

```
/usr/bin/xset s off
/usr/bin/xset -dpms
firefox http://нужный_сайт.com
```

Первые две строчки отключают хранитель экрана и режим сбережения энергии, а затем загружается *Firefox*. Если Вы используете что-то вроде расширения Full Screen для *Firefox*, то браузер будет стартовать в режиме киоска. **НБ**



› **Для полноценной работы Google Earth не обойтись без 3D-ускорения графической карты и драйвера. 2D-ускорение годится только для плоской Земли.**

3 Земля тормозит

У меня есть нетбук Samsung NC10, и я затруднился с работой Google Earth в Ubuntu 9.04. Программа выполняется очень медленно и тратит вечность на изменение масштаба и навигацию по карте. В Windows XP с той же программой проблем нет: все работает хорошо и масштаб изменяется очень быстро. Вдобавок программа прекрасно работает на моей основной машине с Ubuntu 8.10. Искал решение на форумах — без особого успеха. В журналах при запуске программы говорится про /root/.googleearth и /root/googleearth/cache и приводится совет поместить их в каталог /home. На другой моей машине такого не происходит.

Джефф Дингл [Jeff Dingle]

Приложение будет работать с файлами из каталога **/root** только в том случае, если оно запущено от имени **root** или из-под **sudo** в Ubuntu. Это можно делать только с программами, предназначенными для настройки системы.

Обычная причина замедления Google Earth – отсутствие 3D-ускорения в Вашей графической карте или в драйвере для нее. Google Earth настолько зависит от наличия 3D-ускорения, что без него Вы скорее дойдете пешком до той местности, которую хотите увидеть в программе, чем она Вам ее покажет.

Похоже, что в видеодрайвере для карт Intel в Ubuntu 9.04 появились какие-то проблемы, которых не было в более ранних релизах Ubuntu. Первым делом выполните такой тест:

```
glxinfo | grep render
```

и если будет сообщение об использовании программного ускорения – то мы нашли источник проблемы. Причин может быть несколько, как и путей решения, и на страничке Ubuntu Wiki есть их подробное описание. Зайдите по адресу https://wiki.ubuntu.com/X/Troubleshooting/IntelPerformance и разбирайтесь с тамошними рецептами до тех пор, пока не получите 3D-ускорение, достойное Вашей карты. **НБ**

4 Удаление программ

На LXF123 DVD я обнаружил утилиту *ImageKonverter*, которую давно искал. Будучи относительным новичком, я порадовался, что без особых проблем установил и запустил ее. Но, к сожалению, программа толком и не заработала; она распознает только часть моих файлов изображений (а все они — в формате JPEG и скопированы из моей фотокамеры) и иногда не видит ничего в моих папках. Папки *F-Spot* — где вообще-то более 500 фотографий — отображаются пустыми.

Хорошего тут мало, и я попробовал удалить программу. В Установке/Удалении приложений ее нет, нет и в менеджере *Synaptic*, и как мне теперь от нее избавиться? У меня Ubuntu 9.04. Буду благодарен за вашу помощь, поскольку уверен, что этот вопрос всплывет для меня еще не раз.

Дэйв Эйджер [Dave Ager]

Synaptic располагает информацией только о тех пакетах, которые были установлены через него или находятся в доступных ему репозиториях или источниках программ, но ничего не знает о программах, которые Вы собираете из исходных текстов. Установка/удаление приложений использует ту же базу пакетов, так что все вышесказанное применимо и здесь. Многие программы, устанавливаемые с помощью **make install**, можно удалить с помощью **make uninstall**, и *ImageKonverter* – одна из них. Возвратитесь в каталог, в котором Вы запускали **make install**, и запустите

```
sudo make uninstall
```

Если Вы уже удалили распакованное дерево исходников, то снова распакуйте tar-архив с DVD и запустите

```
qmake
sudo make uninstall
```

Необходимости в **make** здесь нет, но может понадобиться **qmake**, поскольку некоторые программы устанавливают файлы по разным путям в зависимости от системы, где происходит установка – 32-битной или 64-битной. Если Вы имеете дело с пакетом, где применяется более распространенные **./configure; make; sudo make install**, то просто выполните

```
./configure
sudo make uninstall
```

Поскольку Вы уверены, что Ваша проблема вернется (и я не стану с вами спорить), то есть еще одна возможность, к которой можно прибегнуть в будущем. *Checkinstall* (http://checkinstall.izto.org) – полезная программа, собирающая пакеты »

из исходных кодов. Установите ее (она есть в репозиториях большинства дистрибутивов), и, дойдя до момента, когда нужно будет запускать **make install**, вместо этого выполните

```
checkinstall -D
```

Checkinstall сама запускает **make install**, наблюдает за тем, какие файлы куда устанавливаются, и затем собирает пакет. Опция **-D** говорит программе, что нужно собрать пакет Deb; также можно создать пакеты RPM и Slackware. Причем пакеты добавляются в базу данных Вашего пакетного менеджера, и позже их можно удалить даже при отсутствии **make uninstall** или аналога, да и незачем хранить исходные коды этой программы.

Хотя *Checkinstall* обычно используют для сборки пакетов с помощью **make install**, она умеет также работать и с другими программами установки, поэтому собрать пакет Deb (или RPM) можно практически для любой программы, которую Вы устанавливаете помимо пакетного менеджера. Таким образом система содержится в полном порядке. **МС**

5 А мой Canon против

В Недавно купил для моей девушки ноутбук Toshiba NB100 с Ubuntu и пытаюсь рассказать ей о преимуществах Linux над Microsoft, преодолевая скептицизм ее братьев! Проблема, которая стоит у меня на пути — принтер Canon Pixma iP100, который не хочет работать в Ubuntu. Скачанные с официального сайта Canon драйверы не заработали из-за какого-то конфликта, и несмотря на наши с подругой усилия, мы никак не можем заставить принтер работать.

Майкл Рекс [Michael Rex]

О Существует пакет с драйверами для Ubuntu, скачать который можно с сайта http://tinyurl.com/lvmwba. Понадобятся два пакета: пакет Debian для iP100 и пакет Debian Common. В одиночку пакет iP100 не установится, что и может быть корнем Вашей проблемы. Первым нужно установить пакет **common**, так что загрузите оба файла в Ваш домашний каталог и для начала дважды щелкните по **cnijfilter-common**, нажмите кнопку Установить пакет и затем повторите процедуру для второго файла. Если Вы попробуете сначала установить **cnijfilter-ip100**, то вместо кнопки Установить пакет увидите сообщение об ошибке. Во время установки может возникнуть сообщение о том, что Вам нужно запустить

```
sudo apt-get install -f
```

в окне терминала. Это означает, что другие файлы, необходимые для этого пакета, не установлены, а вышеприведенная команда найдет их для Вас. Вместо этого также можно проделать все сразу с помощью команды

```
dpkg --install cnijfilter-*
```

Файлы драйвера загрузятся в систему, и Вы теперь сможете пройти по меню Система > Администрирование > Печать [System > Administration > Printing] и настроить Ваш принтер.

Исчерпывающим источником информации обо всем, что относится к принтерам в Linux, является база данных Open Printing на www.openprinting.org, где можно узнать о том, поддерживается ли принтер или нет, а также указано, какой драйвер нужно использовать для принтера.

Альтернатива – Turboprint (www.turboprint.info), набор коммерческих драйверов для принтеров. Есть демо-версия, позволяющая проверить, хорошо ли поддерживается Ваш принтер, прежде чем выкладывать денежки. **НБ**

6 Debian подушкой

В Установил на своем компьютере Debian 5 с диска LXF118. Все работает прекрасно, но я не могу настроить разрешение экрана на правильное значение 1680×1050. Соотношения сторон по умолчанию равны 4:3 или 5:4, а не 8:5, которое нужно для моего ЖК-экрана. Есть ли способ добавить разрешение 1680×1050 в список опций, или я навеки прикован к подушкообразным искажениям?

Дейв Темпл [Dave Temple]

О Вы не указали модель Вашего монитора (хотя с таким разрешением это может быть только очень новая ЖК-модель), но вся линейка более старых CRT-моделей поддерживает стандарт EDID (Extended Display Identification Data, Расширенные данные идентификации дисплея), который позволяет графическим картам получать информацию о возможностях монитора в надежде на то, что X-сервер сумеет подобрать подходящую конфигурацию.

Есть пара возможностей, которые Вы можете испытать; первая из них – переименовать **/etc/X11/xorg.conf** во что-нибудь типа **/etc/X11/xorg.conf.old** и затем перезапустить X. В отсутствие **xorg.conf** свежая версия *X.Org* настроит дисплей на лету, для чего ей придется прочесть информацию о возможностях монитора.

Также можно заставить Debian перенастроить X.Org-сервер таким манером:

```
dpkg-reconfigure xserver-xorg
```

Проверить, выдает ли графическая карта правильную EDID-информацию, можно, введя от имени root

```
get-edid | parse-edid
```

Команда **get-edid** считывает информацию с монитора, а **parse-edid** переводит ее в человеко-читаемый формат, который также можно использовать и в **xorg.conf**.

Если все это не даст результатов, можно отредактировать **/etc/xorg.conf** и добавить правильные видеорежимы вручную. Изменения делаются в секцию **Screen**, где по умолчанию обычно пишется нечто вроде следующего:

```
Section "Screen"
 Identifier "Default Screen"
 Monitor "Configured Monitor"
EndSection
```

После внесения изменений она должна выглядеть примерно так:

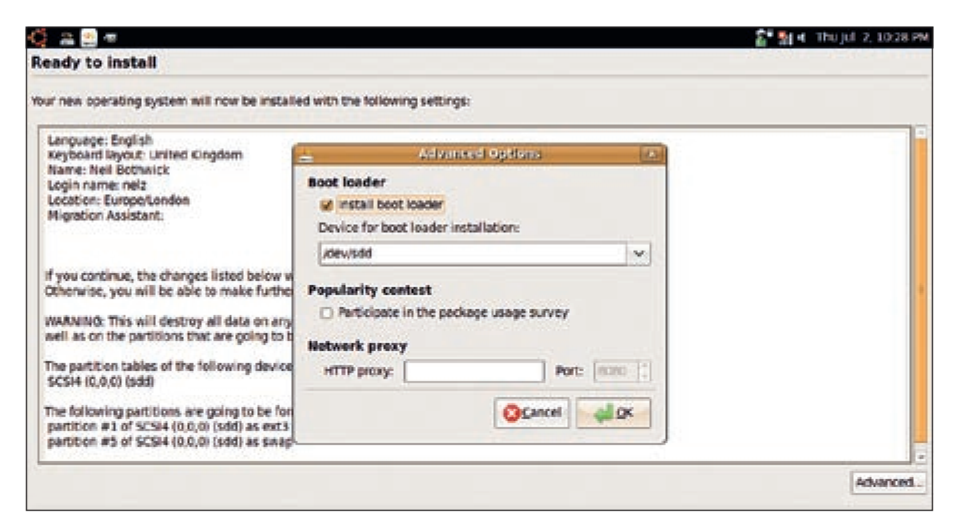
```
Section "Screen"
 Identifier "Default Screen"
 Monitor "Configured Monitor"
 DefaultDepth 24
 SubSection "Display"
  Depth 24
  Modes "1680x1050" "1280x800" "1024x768"
 EndSubsection
EndSection
```

Для каждого значения глубины цвета экрана понадобится один подраздел **SubSection** (хотя, вероятно, Вам понадобится всего один подраздел для глубины в 24 бита). Внутри этого подраздела укажите возможные видеорежимы. Порядок важен, поскольку первый режим будет использоваться как режим по умолчанию. Добавленные Вами разрешения должны стать доступными в настройках Параметры экрана [Screen Resolution] после перезагрузки X. Для этого нажмите Ctrl+Alt+F1, чтобы попасть в виртуальный терминал (окно X-терминала, такого, как Gnome Terminal, здесь не годится), авторизуйтесь как root и запустите

```
/etc/init.d/gdm stop
/etc/init.d/gdm start
```

Если Вы используете KDE, а не Gnome, замените **gdm** на **kdm**. Также можно заменить отдельные команды **stop** и **start** на одну **restart**, но это сработает не во всех дистрибутивах. **НБ**

› Будет ли ваш струйник печатать или просто служить подставкой для бумаг? На OpenPrinting.org собрано все, что относится к принтерам в Linux.



› Устанавливая дистрибутив на сменное устройство, поместите туда и загрузчик, иначе при неподключенном устройстве система откажется запуститься.

7 Незагружаемый ноутбук

В Я давно прикидываю, не поработать ли в Linux, особенно в последнее время, так как Vista мне ненавистна, а поддержка XP скоро закончится. В прошлом году я серьезно взялся было за OpenSUSE, но только, что называется, пальцы обжег. Мое любопытство снова разгорелось, когда я увидел LXF123 и решил купить журнал. Найдя в рубрике «Ответы» инструкцию по установке на USB-флэшку, я снова «вспыхнул», поскольку появился шанс все попробовать по-настоящему, не меняя ничего на ноутбуке.

Сверяясь с журнальной инструкцией, я загрузил ноутбук с Ubuntu Live CD и поставил систему целиком на USB-флэшку объемом в 16 ГБ. Установка прошла без сучка и без задоринки (по крайней мере, мне так показалось). В начале загрузки я нажал F12, изменил меню загрузки и загрузился затем в Ubuntu. Да, работало все не очень быстро, но работало отлично. Беспроводная сеть настроилась, я мог слушать MP3, получил доступ к файлам на моем домашнем компьютере (Windows!) и мог редактировать документы.

А затем возникла проблема. Я включил ноутбук без флэшки, поскольку мне понадобилась Windows. Появился загрузочный экран *Grub* с сообщением об ошибке, без всяких указаний на возможность продолжать загрузку. Намертво! Выходит, что хотя я и использовал CD и установил систему прямиком на флэшку, *Grub* все-таки окопался на жестком диске, и теперь ноутбук загружается только с флэшкой, а без нее — ни в какую. Хорошо еще, что я не успел использовать флэшку под что-то другое!

Тревор Диппер [Trevor Dipper]

О Установка загрузчика проходит в два этапа. Большая часть кода, включая файл настроек загрузочного меню, заносится в каталог **/boot** устанавливаемой Linux-системы. Начальный код загрузчика, который передает контроль над процессом этим файлам, записывается в главную загрузочную запись (Master Boot Record, MBR) диска; там же содержится таблица разделов. Куда бы Вы ни установили файлы *Grub*, установщик по умолчанию разместит код MBR на первый загрузочный диск (как правило, это Ваш жесткий диск), потому что иначе меню *Grub* никогда не появится при нормальном процессе загрузки. В ответе в LXF123 особо оговаривалось, что Вы должны нажать кнопку Advanced [Дополнительно] и изменить место, в которое будет установлен загрузчик, на карту памяти (это кроме указания ее в качестве «диска», на который будет производиться установка). Если проделать все именно так, то уже имеющийся на жестком диске загрузчик остается нетронутым, и компьютер будет загружаться прямо в ранее установленную ОС, как и всегда. Теперь для получения доступа к загрузочному меню *Grub* Вам нужно, чтобы флэшка была заранее вставлена.

Если разместить *Grub* в MBR жесткого диска, то он будет жаловаться на отсутствие остальных »

Часто задаваемые вопросы

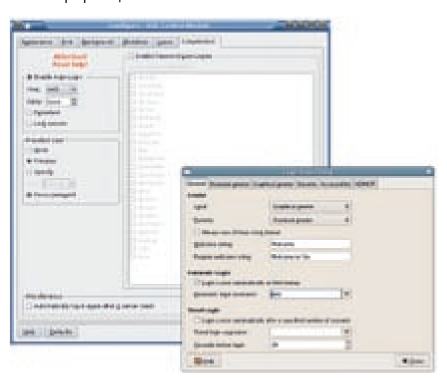
Пользователи и суперпользователи

» Я только что впервые установил Linux. Зачем нужно было создавать двух пользователей?
Одним из этих двух пользователей был root, или суперпользователь, который выполняет только административные задачи. Другой же – обычный пользователь для повседневной работы в системе.

» А нельзя администрировать систему самому?
В Linux используется набор разрешений, которые устанавливают права владения файлами и каталогами и доступа на чтение и запись в них. У каждого пользователя есть свой домашний каталог, где он волен делать с файлами все, что взбредет ему в голову: хоть создавать, хоть удалять. В чужих каталогах пользователи могут читать файлы и запускать программы, но не изменять или удалять их, что обеспечивает надежность и безопасность. Нечаянно удалить системные файлы невозможно, и даже запуск зловредного вложения в почту не испортит систему, поскольку у вас нет прав, необходимых для совершения вредоносных действий. Когда вам действительно нужен доступ с правами **root**, скажем, для установки программы, вы переключаетесь на пользователя **root** до того момента, пока работа не будет сделана.

» Выходит, мне нужно выйти вон и войти как root при каждой установке программы. Не многовато ли мороки?
Нет, вам не придется заканчивать сеанс, и тем более не придется запускать целый рабочий стол для **root** (в некоторых дистрибутивах это даже не допускается). Нужно только выполнить с правами root одну конкретную программу – во многих случаях такая возможность уже настроена. При попытке запустить *Yast* – программу настройки SUSE – или Центр управления Mandriva система запросит у вас пароль root. Для любой другой программы откройте терминал, введите **su** и пароль и затем запускайте программу, введя ее название.

» Моим компьютером я пользуюсь один; зачем мне каждый раз проходить авторизацию?
Если вы единственный пользователь своего компьютера, то в большинстве дистрибутивов есть удобная возможность автоматического входа для указанного пользователя (не root!). В KDE это можно настроить из раздела настроек Системное администрирование > Менеджер входа в систему [System Administration > Login Manager]. Пользователям Gnome надо будет открыть меню Система > Администрирование > Окно входа в систему [Desktop > System Settings > Login Screen] из панели меню и отметить опцию Включить автоматический вход в систему вкладки Безопасность [Automatic Login во вкладке General]. Но дважды подумайте, прежде чем задействовать эту возможность на ноутбуке: если он потеряется или его украдут, то любой, кто его включит, получит доступ к вашей почте и другой личной информации.



› И KDE, и Gnome допускают автоматический вход для пользователя при загрузке системы.

файлов – что у Вас и произошло. Но решение существует. Загрузитесь со спасательной дискеты или раздела Windows, войдите в восстановительную консоль Windows и запустите **fixmbr**. Загрузчик Windows восстановится.

Если в Вашей Windows нет такой возможности, то восстановить загрузчик Windows может Super Grub Disk с сайта www.supergrubdisk.org. **МС**

8 Solaris не сдается

В LXF122 появился LiveCD OpenSolaris 2009.06. Это замечательно — я записал образ на диск и загрузился с него. Однако в статье, посвященной данной ОС, сказано, что после загрузки мы попадаем в Gnome. На самом деле в Gnome мы не попадаем, и до запуска графической подсистемы дело вообще не доходит: вместо этого предлагается ввести имя пользователя и пароль в консоли! Какие реквизиты требуются — не сказано ни в статье, ни на прилагаемом к журналу диске. Я попробовал самые разные ходовые варианты, но безуспешно. Может быть, вы что-то упустили при формировании образа? Хотелось бы все же поработать с OpenSolaris.

Виктор Басынин

Образ OpenSolaris, который был представлен на **LXF**DVD, совершенно стандартный: если кто-то что-то и упустил при его сборке, то это были инженеры Sun Microsystems. Не то чтобы они (равно как и мы) были безгрешны и никогда не делали ошибок, но в данном случае причина кроется в другом. Как Вы справедливо заметили, графическая подсистема не стартует, а это явным образом указывает на проблемы совместимости с оборудованием. Менеджер дисплея GDM, используемый на LiveCD OpenSolaris, настроен на автоматический вход в систему, поэтому при нормальном раскладе никакой пароль вводить не требуется. Но поскольку запуск GDM, по-видимому, завершается неудачей, система пытается продолжить загрузку в текстовом режиме, который ничего не знает про автовход. Имя пользователя и пароль, настроенные на LiveCD – jack/jack; как можно видеть, даже реквизиты из четырех букв не всегда поддаются лобовой атаке. Они позволят Вам зарегистрироваться в системе; если потребуются привилегии root, его пароль – opensolaris. **ВС**

9 Пустите в Сеть!

В Помогите, пожалуйста, настроить Интернет через VPN-соединение. Дистрибутив — OpenSUSE 11.1, рабочий стол — GNOME, делал все по инструкции с http://ru.opensuse.org/VPN — не заработало. Тогда попробовал настроить подключение через *NetworkManager 0.7*, установил подключаемый модуль *NetworkManager-pptp* — и тоже безрезультатно.

Есть предположение, что причина в типе VPN, так как провайдер требует выбирать L2TP (разумеется, инструкция есть только для Windows), а *NetworkManager* предоставляет PPTP.

Если можно, опишите, пожалуйста, подробно процесс настройки. Или подскажите какой-нибудь другой путь подключения к Интернету.

NKS H

Предположение Ваше абсолютно верное – к сожалению, поддержка L2TP в *NetworkManager 0.7* еще не реализована. Не могу сказать, значится ли она в планах разработчиков, но настроить L2TP-канал можно и без нее. Для этого существуют два способа, но для начала удалите *NetworkManager* и сопутствующие пакеты: они могут помешать.

Во-первых, Вы можете воспользоваться графической утилитой *Kvpnc* (home.gna.org/kvpnc); она поддерживает множество VPN-протоколов, включая L2TP. В разделе Download официального сайта вы найдете готовый пакет для OpenSUSE (архитектуры i586 и x86_64). Главный недостаток *Kvpnc* в Вашей ситуации в том, что это – приложение KDE, а не Gnome. Если Вас смущает данный факт, можно настроить соединение вручную. Установите пакет *xl2tpd* и создайте (от имени root) файл **/etc/xl2tpd/xl2tpd.conf** примерно такого вида:

```
[global]
access control = yes
[lac provider]
lns = IP-адрес сервера провайдера
require authentication = no
pppoptfile = /etc/ppp/options.xl2tpd
```

Содержимое **/etc/ppp/options.xl2tpd** должно напоминать следующее (правда, оно сильно зависит от настроек на стороне провайдера, и не исключено, что Вам придется прибегнуть к методу проб и ошибок):

```
lock
noauth
nobsdcomp
nodeflate
noaccomp
nomppe
defaultroute
name ваше_имя
password ваш_пароль
```

Основными здесь являются имя и пароль, а также строчки с **no**, которые отключают сжатие и шифрование (как правило, провайдеры не используют ни то, ни другое: VPN нужен им только как средство авторизации). **Defaultroute** добавляет при подключении маршрут по умолчанию. Далее, наберите как root

```
xl2tpd
```

чтобы запустить сервер. Если все пройдет гладко, соединение будет установлено. Если нет – попробуйте добавить в **xl2tpd.conf** строку **ppp debug=yes** и понаблюдать за журнальными файлами. Распространенной проблемой бывает нахождение L2TP-сервера провайдера в собственной подсети (скажем, Вы – 10.1.0.15, а сервер – 1.2.3.4) – в таком случае не забудьте прописать до него прямой маршрут через Вашу сетевую карту. **ВС**



Коротко про...

Компиляцию программ

Большинство дистрибутивов располагают обширными репозиториями пакетов почти для всех программ, которые вам могут потребоваться, но иногда бывает нужно собрать приложение из исходных текстов. Типичные причины – для вашего дистрибутива нет пакета с новой версией программы, или же вы хотите наложить заплатку на исходные тексты, чтобы добавить опцию или исправить ошибку. Всегда предпочтительнее использовать менеджер пакетов вашего дистрибутива, но если все-таки без сборки из исходников не обойтись, то последовательность тут простая.

Для начала распакуем tar-архив (или «тарбол»), одной из двух команд:

```
tar xvzf foo-1.2.3.tar.gz
tar xvjf foo-1.2.3.tar.bz2
```

Обычно исходники помещаются в каталог с тем же именем, что и архив; зайдите в него командой **cd foo-1.2.3**. Найдите файлы с названием **README** или **INSTALL** и прочтите их. В них обычно содержатся инструкции по установке. Типовая инструкция такова:

```
./configure
make
su -c "make install"
```

Первая команда проверяет систему, убеждается в наличии нужных зависимостей и настраивает все возможные опции программы. Неплохо также вначале запустить **./configure --help**, чтобы увидеть доступные опции.

Вторая команда проводит компиляцию, помещая создаваемые файлы в текущий каталог. Чтобы сконфигурировать или собрать программу, не обязательно иметь права root, но третий шаг – копирование собранных файлов в системные каталоги, и тут эти права понадобятся, поэтому мы и прибегаем к **su**, чтобы запустить из-под root только одну эту команду. Пользователи Ubuntu должны заменить ее на

```
sudo make install
```

Собранная программа, как правило, устанавливается в каталог **/usr/local/bin**. Если вы работаете в системе на базе RPM, и **configure** жалуется на отсутствующую библиотеку, а вы точно знаете, что она есть в системе – тогда установите соответствующий devel-пакет, например, **libbar-devel**.

Большой вопрос Как починить кривую установку Fedora?

10 Федорино горе

В Я попробовал live-версию Fedora 11, и все работало нормально, но после установки ее на второй жесткий диск обнаружились проблемы. Аудиодиски не определялись или не показывались на рабочем столе, и ни *Rhythmbox*, ни *Exaile* их не проигрывали, хотя с *Banshee* все было нормально. Мой USB-ТВ-тюнер DVB-t не определился, и я не могу смотреть ТВ в *Kaffeine*.

Два моих жестких диска не координировались друг с другом при загрузке; Fedora 11 загружалась, но ранее установленных дистрибутивов не было в загрузочном меню. Похоже, что по умолчанию создалось два раздела — загрузочный в 200 МБ и второй — LVM-раздел, про который *GParted* сказал, что пока его не поддерживает. Раздела подкачки я не увидел, хотя он мог быть в разделе LVM.

Бернард Ривз [Bernard Reeves]

О Похоже, что в Fedora налицо конфликт аудиодисков с конкретной версией *udev*, что можно исправить, установив более старую версию. Запустите команду

```
yum downgrade udev libudev0 libvolume_id
```

от имени root. Только учтите, что на момент выхода журнала проблему могли уже решить; тогда Вы отделаетесь обновлением пакета. Способ доступа к аудиодискам у каждой программы свой, поэтому одни работают, невзирая на проблему, а другие – нет. *VLC* (www.videolan.org/vlc) также работает, хотя *udev и* испорчен.

Некоторым устройствам USB DVB для работы нужен файл аппаратной прошивки. Поскольку драйверы являются частью ядра, эти двоичные файлы не могут поставляться в его составе, и устанавливать их надо отдельно. В зависимости от сборки и модели Вашего устройства, пакет для прошивки можно найти в Установке/удалении приложений, но в большинстве случаев файл нужно будет скачать с www.linuxtv.org. Это не трудно – просто запустите **dmesg** после подключения устройства, для его определения, затем получите соответствующий файл и скопируйте его в каталог **/lib/firmware**. Если Вы не уверены, какой именно файл Вам нужен, спишите в этот каталог все подряд. Теперь – ко второй части Вашего вопроса...

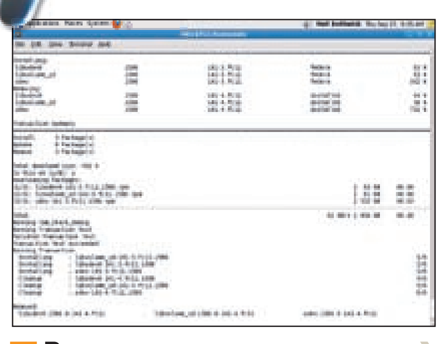
Инсталляторы всех основных дистрибутивов находят установку Windows и настраивают меню для двойной загрузки, но не все делают это для своих установленных Linux-собратьев. Простейший способ исправить положение – настроить меню Fedora на вызов загрузочных меню каждого из установленных ранее дистрибутивов, для чего отредактируйте **/boot/grub/grub.conf** из-под root в Вашей Fedora, добавив такие строчки:

```
title Ubuntu
root (hdX,Y)
chainloader +1
```

Строка, содержащая **root**, сообщает, откуда производить загрузку; **X** и **Y** здесь – номера диска и раздела соответственно. *Grub* начинает отсчет с нуля, поэтому первый раздел второго диска будет называться **(hd1,0)**. Повторите процедуру для Mint, и получите загрузочное меню, где будут все установленные у Вас дистрибутивы.

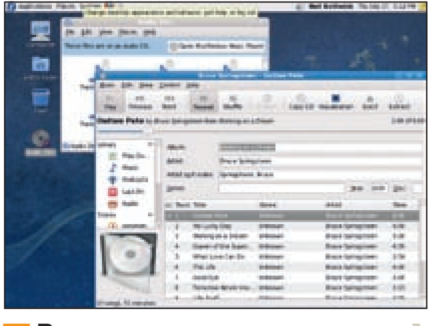
По умолчанию Fedora применяет Logical Volume Manager (LVM), который работает с маленьким загрузочным разделом, где расположено ядро и файлы, необходимые для инициализации логических томов, включая файловую систему раздела подкачки. *GParted* не станет с ним работать, потому что он предназначен только для физических разделов.

Шаг за шагом: Наладим Grub и проигрывание CD



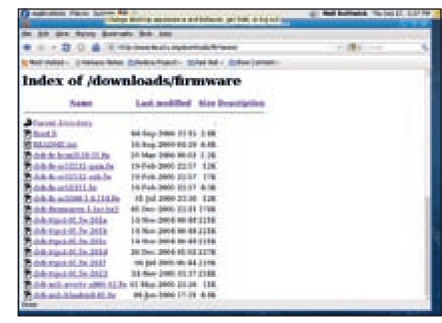
1 Выполним «устарение»

Запустив *Yum* из командной строки, можно заменить версии программ, таких, как *udev*, который не дает нормально проигрывать аудиодиски, на более старые.



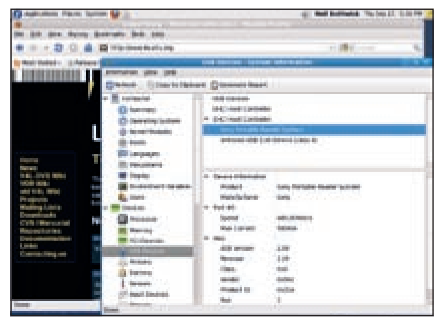
2 Возвратим аудио-диски

После установки более старой версии пакета *udev* и перезагрузки, аудио-CD должны показываться на рабочем столе и в *Rhythmbox*.



3 Получим прошивку

На сайте www.linuxtv.org есть файлы прошивок для различных TV-устройств, их нужно разместить в **/lib/firmware**.



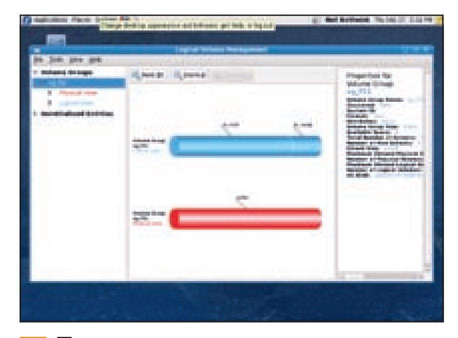
4 Что там в коробочке?

Установите программу *Hardinfo*, чтобы иметь подробную информацию об аппаратных составляющих вашего компьютера. Это поможет при поиске драйверов и прошивок.



5 Grub — каждому

Добавление дополнительных пунктов в файле меню *Grub* даст возможность загружать другие дистрибутивы, которые вы установили ранее.



6 Логические тома

Fedora хранит все, кроме раздела **/boot**, на логических томах, включая и раздел подкачки. «Управление логическими томами» покажет вам подробности. LXF

LXFHotPicks



Энди Хадсон
Когда Энди не притворяется, что отлаживает почтовые сети, он ныряет среди коралловых рифов в поисках жемчужин для HotPicks.

Bilbo » ClamTK » FinalPage » Areca » BMTG3 » CloneRNX » Choqok » TuxPaint » KDESvn » KAlternatives

Инструмент для создания блогов

Bilbo

Версия 1.0 **Сайт** http://bilbo.gnufolks.org

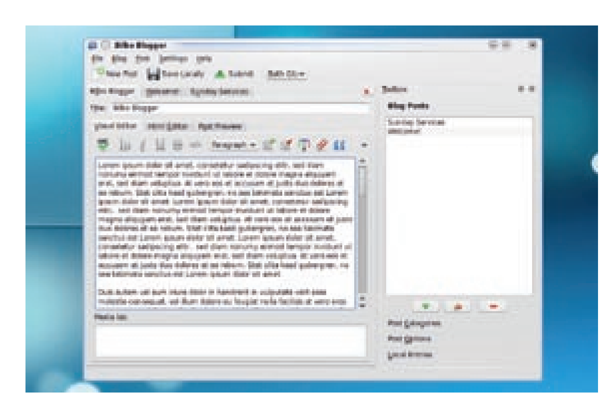
Хвалите Господа за современные платформы для блоггинга. С ними вы сможете организовать роскошный блог менее чем за 30 минут. Впрочем, создание сайта – еще полдела: только интересное и постоянно обновляемое содержание гарантирует вам неиссякающий приток посетителей.

Однако необходимость ввода пароля при каждой попытке обновить свой блог может сильно достать; так почему бы не воспользоваться клиентом, помогающим создавать новые посты и поддерживать уже опубликованные? *Bilbo* – один из таких, и он только что вышел в версии 1.0. Он основан на KDE, отлично смотрится в KDE 4.3 и поддерживает множество платформ, в том числе *WordPress*, *Blogger* и *Movable Type*.

Настройка чрезвычайно проста: укажите *Bilbo* основной URL своего блога и введите свои имя и пароль. *Bilbo* попытается подобрать настройки автоматически, основываясь на ответах на несколько запросов. При необходимости вы сможете отрихтовать эти параметры, но они и так должны получиться удачными.

Одно соберет их

Запустите *Bilbo*, и вы увидите опрятный интерфейс с основным редактором, размещенным в центре экрана, и 20 самых последних постов, которые автоматически выводятся справа. Чтобы просмотреть содержимое любой записи, дважды щелкните на ней – это позволит отредактировать или обновить его при необходимости. При открытии постов для редактирования *Bilbo* создаст новую вкладку для каждого, и у вас будет в наличии сразу несколько для перекрестных ссылок. Некоторые платформы блоггинга позволяют снабжать записи тэгами и категориями, что весьма полезно, если вы аккуратист и хотите, чтобы в ваших постах все было тщательно организовано. На следующий уровень вас переводит поддержка *WordPress*: она позволяет создавать постоянные ссылки [permalinks], чтобы адреса ваших постов было проще запомнить.



› *Bilbo* облегчает задачу обновления блога.

Главное окно редактирования тоже богато функциями. Имеется встроенная поддержка нескольких блогов с простым выпадающим меню, позволяющим быстро и без мороки переключаться между ними. Завершает впечатляющий список функций возможность работы оффлайн: при этом посты ваших блогов сохраняются локально, и вы сможете загрузить их при следующем выходе в сеть. Если соединение с сетью для вас большая проблема, это может стать жизненно важным.

По части зависимостей, перед началом работы нужно убедиться, что у вас установлены KDE 4.x и *libboost*. Компиляция достаточно проста, и весьма приятно видеть, что, несмотря на относительную молодость, данный проект уже попал в пакеты ряда популярных дистрибутивов. По нашему мнению, это – одно из классных приложений KDE, и оно явно стоит вашего внимания.

Исследуем интерфейс Bilbo

Трибуна
Нажмите здесь, чтобы создать новый пост, и готовьтесь высказать свои сокровенные мысли.

Спаси и сохрани
При работе оффлайн, жмите на эту кнопку, чтобы сохранить свой пост локально на то время, пока не отыщется сетевое соединение.

Форматирование
Все, чего и следовало ожидать, найдется здесь, вместе с опцией удаления всего форматирования.

Рулит контент
Именно здесь вы будете хранить свои перлы мудрости, чтобы открыть их миру.

Список мультимедиа
Если вместе со своим постом вы загружаете мультимедиа, здесь появится его список для подтверждения.

Блог-посты
Нажмите здесь, и отобразятся 20 самых свежих постов, внесенных вами в блог.

Всегда на связи
Создайте постоянную ссылку на свой блог, используя возможности *WordPress*.

Антивирусный сканер

ClamTk

Версия 4.17 **Сайт** http://clamtk.sf.net

Много лет пользуясь Linux, мы знаем, что шанс подцепить вирус весьма невелик. Однако дистрибутив все же стоит обезопасить, поскольку вам, возможно, случается сотрудничать с друзьями, коллегами или членами семьи, предпочитающими Windows.

Дело в том, что хотя сам по себе Linux невосприимчив к вирусам Windows, он может стать бациллоносителем: файлы, содержащие вирусы, «забредают» в Linux, а потом перекочевывают к нашим более уязвимым братьям. Можно, конечно, считать это уже не нашей проблемой, но мы полагаем, что некая толика компьютерной гигиены отнюдь не повредит.

ClamTk, имеющийся в пакетах для разных дистрибутивов вместе с исходным кодом, предлагает куда более простой интерфейс, чем рассмотренный в **LXF92**. Удобное окно предоставляет важную информацию о состоянии *ClamAV* (поверх которого все и построено), а также о том, когда в последний раз был обнаружен вирус. Над панелью Status наблюдается целая серия флажков, позволяющих уточнить критерии сканирования, а также три дополнительных кнопки для выбора областей системы, которые вы хотите проанализировать.

Тест на заразу

Настройка сканирования – проще не бывает: просто выберите опции, которые вы хотите включить в процесс – например, скрытые файлы и директории – и нажмите подходящую кнопку на панели инструментов Action. Нажатием на кнопку Home вы вполне предсказуемо ограничите сканирование своей домашней директорией, но можно также указать отдельный файл или каталог (и его подкаталоги) для специальной обработки. После этого *ClamTk* примется за работу, и под областью Status появится строка состояния, где будет отображаться количество проверенных файлов и обнаруженных вирусов.

> «ClamTk отлично справится с поддержанием чистоты.»

Найдя вирус, *ClamTk* уведомит вас об этом и поместит инфицированные файлы в карантин, где они бессильны причинять вред. Причем вам не понадобится много зависимостей. Если вы решите компилировать пакет из исходников, просто убедитесь, что у вас установлены *ClamAV* и *perl-Gtk2*.

Короче говоря, в тех редких случаях, когда он может вам понадобиться, *ClamTk* отлично справится с задачей поддержания чистоты в вашем компьютере.



› **Пусть ваши друзья будут счастливы, а их система — свободна от вирусов.**

Работа с PDF

Final Page

Версия 0.3.0 **Сайт** http://finalpage.sourceforge.net

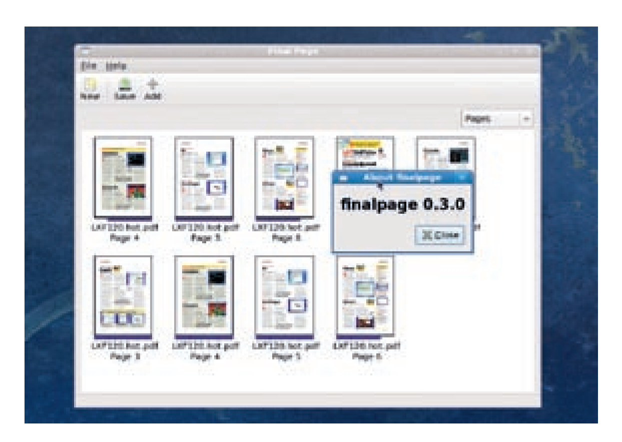
PDF стал исключительно популярным форматом файлов. Если вам нужен документ, пригодный для чтения практически во всех основных системах, именно PDF сохранит и содержимое, и форматирование. Программы для работы с PDF имеются в изобилии, и разнообразные приложения способны и создавать, и открывать файлы. Однако, конвертировав файл в PDF, вы, как правило, уже мало что можете с ним сделать, если, конечно, не жаждете посвятить себя поискам программ для извлечения содержимого из этого формата.

Ну, то есть, если только вы не дорвались до *Final Page*, который запросто расчленяет файл PDF на более мелкие части уже после создания. Откройте PDF через *Final Page*, и вы сможете просмотреть его одним из двух способов. Во-первых, можно выбрать просмотр документа в виде набора страниц. Может, и не Бог весть какое откровение, но зато вы отследите точное количество страниц, с которыми в данный момент работаете. Очень полезны синие маркеры под каждой страницей, позволяющие увидеть, какие страницы сгруппированы вместе; а маркеры начала и конца четко определяют границы каждого просматриваемого документа.

Прекрасный вид

Еще один способ отображения – в виде документа, и опять же ничего нового в этом нет; но стоит вам начать разделять страницы на отдельные PDF, как это поможет отслеживать листы, формирующие новые документы. Интерфейс достаточно прост, он позволяет разделять документы, вращать страницы и изменять их порядок. Сохранение происходит очень быстро, но допускается сохранять только один открытый документ за один раз, так что ненужные страницы или документы придется удалить.

Несколько удивляет обилие зависимостей – для успешной компиляции *Final Page* необходимы *gdk-pixbuf*, *gee*, *poppler-glib* и *gtk+unix-print*. Учтите, что на данный момент вы не сможете объединять PDF, но разработчик пообещал, что в следующей версии это появится. В общем и целом, *Final Page* – удобный инструмент, и мы с нетерпением ждем будущих версий и усовершенствований.



› **Разделяйте и властвуйте над своими PDF с помощью *Final Page*.**

Утилита резервного копирования

Areca

Версия 7.1.5 **Сайт** http://areca-backup.org

У каждого системного администратора есть наготове своя сага о резервном копировании. Сюжет расхожий: как нечто сработало нештатным образом и сколько героики потребовало спасение ценных данных. К счастью, мы, пользователи Linux, имеем достойное оснащение для резервного копирования; моментальные копии данных и файлов делаются легко, с помощью инструментов командной строки и плановых задач *Cron*.

Однако было бы ошибкой ограничиваться только этими средствами, коль скоро имеются программы, способствующие созданию эффективного режима резервного копирования. Мы уже рассматривали некоторые из них, но как-то не замечали *Areca Backup* – и очень жаль.

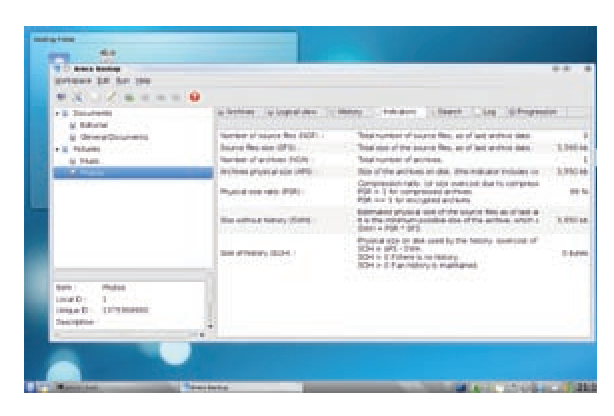
Она основана на Java и является кроссплатформенной, что идеально при работе в нескольких средах. Изначальный интерфейс не слишком продвинут – вам надо создать новую цель, чтобы начать резервное копирование чего бы то ни было. Как часть своего резервного копирования, вы можете создавать множественные цели, чтобы охватить разные области файловой системы, и создав новую цель, можно выбрать сохранение данных в локальном репозитории (разрешается монтировать NFS). Альтернатива – выгрузка резервных копий на FTP-сайт; *Areca* допускает использование Secure FTP, что должно примирить вас с передачей своих данных через общественную сеть. Вместо использования традиционного мастера настройки цели, *Areca* предоставляет список заголовков, из которых необходимо сделать выбор. Это не столь интуитивно, как могло бы быть, но большой беды тут нет.

Выберите источник

Вы можете организовать сохранение данных в виде полной копии источника или изменений, или сохранить данные внутри образа, что может весьма облегчить работу с файлами резервных копий. Допускаются множественные источники – здесь-то интерфейс *Areca* и докажет свою эффективность: он позволяет перетаскивать папки и файлы из файлового менеджера в выбранное окно, ускоряя процесс.

Отлично работает сжатие, дающее возможность использовать zip или zip64, а также хранить все скопированные файлы в одном большом архиве или разделить его на разные файлы. Резервные копии можно распределить по нескольким ZIP-файлам, задав верхний предел размера ZIP-архива.

Предусмотрено и шифрование, с вариантами использования собственной кодовой фразы (индикатор покажет силу пароля) или ее генерации случайным образом в *Areca*. В том же окне, где производится шифрование, находятся управление файлами и опции обхода: *Areca* умеет заглядывать в поддиректории и даже следовать символьным ссылкам. Можно также настроить *Areca* на игнорирование пустых директорий и учет прав доступа к файлам.



› *Areca* **берет на себя немалую часть тяжкой работы по настройке режима резервного копирования.**

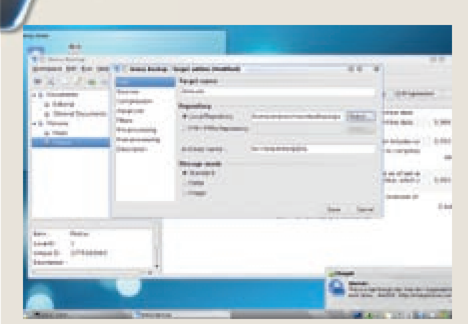
> «Интерфейс позволяет перетаскивать папки и файлы в окно.»

Стратегические планы

Хотя *Areca* и является инструментом Java, она способна изготовлять скрипты оболочки, годные для включения в crontab, что позволяет взнуздать мощный мастер планировки: он создает стратегию резервного копирования на основе предоставленной вами информации. Имеется также опция создания разовых задач и уточнения типа резервного копирования – это идеально, если вам нужно наскоро сделать инкрементное резервирование. Интерфейс просто спартанский, и, в отличие от других программ, здесь нет множества кнопок, засоряющих галереи панелей инструментов.

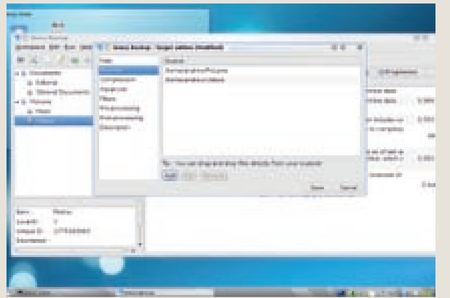
И, наконец, стоит упомянуть, что наследие Java в *Areca* весьма упрощает ее настройку. Если вас не испугает слегка неэффективный интерфейс, она стоит вашего внимания и усилий.

Шаг за шагом: Настройка резервных копий



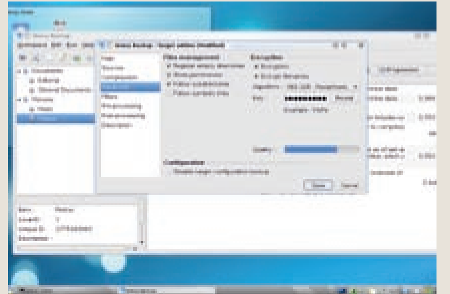
» Выбор цели

Первым делом укажите *Areca*, где вы хотите создать резервную копию, введите имя архива и способ сохранения.



» Задание директорий

Затем укажите копируемые директории и файлы. Вы можете либо искать их, либо перетащить в окно Sources.



» Никому не скажу

Имеется возможность зашифровать свои резервные копии, вплоть до шифрования имен файлов. Главное – не забыть пароль!

HotGames Развлекательные приложения

Головоломка

BMTG3

Версия 0.3.1 **Сайт** www.alarmred.com/node/29

Стратегии реального времени, вроде *Dune 2000* и *Command & Conquer*, всегда приносили массу удовольствия. Хотя условия просты – создавайте боевые единицы, способные помочь вам победить противника – в стратегии все вращается около создания правильных родов войск для достижения поставленных целей.

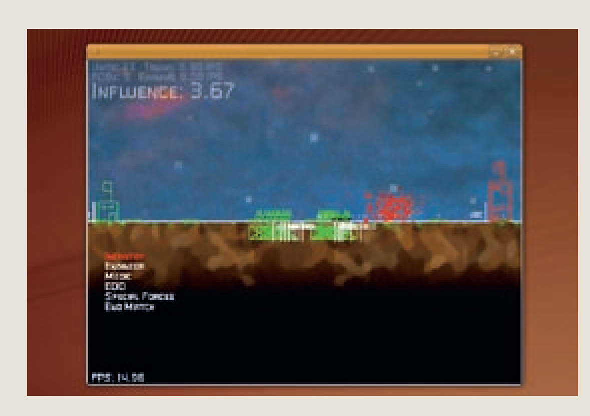
Blood Makes the Grass Grow Green [На политой кровью земле трава зеленее] (она же – *BMTG3*) добавляет в эту формулу свои составляющие: действие разворачивается на горизонтальной плоскости, вы и ваш враг занимаете противоположные стороны поля боя. Вокруг разбросаны здания, которые следует захватить. Но ваши батальоны могут перемещаться лишь в одном направлении, и их нельзя остановить мгновенно или заставить отступить, так что время решает все.

Свершать завоевания вам помогают пехотинцы, опытные саперы, специальные группы диверсантов и многое другое. Чем больше строений вы удерживаете, тем быстрее растет ваше влияние, позволяя вам создавать свои армии с большей скоростью. Освоившись с основной идеей, направьте все свои силы на борьбу с AI в режиме одного игрока или сыграйте против другого человека.

Поляны травой зарастут

Играя, вы обнаружите, что на местах гибели людей начинает пробиваться трава, что, видимо, и обусловило столь длинное название игры. Игровые объекты намечены лишь в общих чертах, поэтому игра не кажется чересчур кровожадной, а лаконичная графика сейчас в моде. Она не отвлекает от самой игры и позволяет *BMTG3* работать невероятно плавно, в среднем – на отметке около 30 кадров в секунду.

> «На местах гибели людей начинает расти трава.»



› **Поле битвы залито кровью — ну, то есть, красными пикселями.**

Что касается зависимостей, первое требование – SDL, что в наши дни стало вполне обычным, а также нужны *SDL_Mixer*, *Image* и поддержка TTF. Позаботившись обо всем этом, запустите *BMTG3* командой **launch** и принимайтесь поливать своего неприятеля свинцовым дождичком.

Головоломка

CloneRNX



Версия 1.0 **Сайт** www.sauronsoftware.it/projects/clonernx

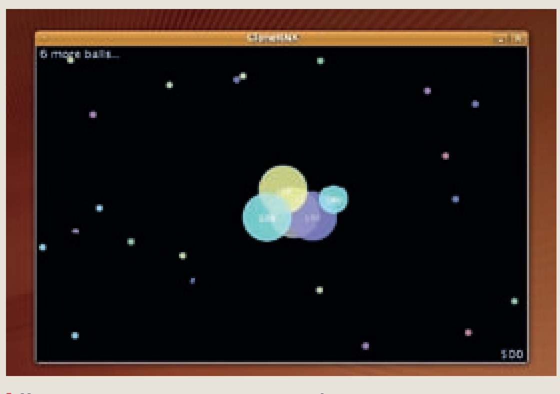
Нам уже случалось отмечать, что самые лучшие игры основаны на простейших концепциях, и *CloneRNX* явно подпадает под это определение. Цель игры – особо не мудрствуя, уничтожать серии шаров и переходить на следующие уровни.

Вообще-то все сложнее, чем может показаться; шары возникают случайным образом в самых разных местах экрана, а для уничтожения первого шара вы вооружены единственным зарядом взрывчатки. Лопаясь, шар провоцирует еще один взрыв, и есть надежда на возникновение цепной реакции, но взрывы весьма кратковременны. За каждый взорванный шар начисляются очки, а за каждый дочерний взрыв вы получаете бонус, который поможет вам достичь высоких результатов.

Чтобы натренироваться, потребуется не так уж много практики, и никаких особо сложных правил, куда нацелить начальный удар, нет. Логика подсказывает, что лучше всего направить его в центр, но это срабатывает не всегда.

К счастью, первые уровни очень легки, и для продвижения вперед нужно всего лишь взорвать несколько шаров. Однако по мере продвижения приходится уничтожать их все больше. На конечном, 12-м уровне, вам придется поразить 54 из 60 шаров. Когда вы это сделаете, победное чувство будет безмерным, способным сравниться лишь с испытываемым при «адмиральском» завершении комбинации в домино. Прямо-таки жаль, что разработчики пока не написали больше уровней.

> «Сперва вы вооружены единственным зарядом взрывчатки.»



› **На вид стохастично, но поверьте: в финале приходит чувство внутренней умиротворенности.**

Запустить *CloneRNX* легко – просто выполните имеющийся JAR-файл Java. А значит, вы без проблем сможете передать ее друзьям, использующим другие системы, если захотите отнять у них пару часов.

Клиент микроблогов

Choqok

Версия 0.6.6 **Сайт** http://choqok.gnufolks.org

Мы не таим своей любви к Twitter, платформе микроблогов, которая позволяет миллионам людей делиться с другими мгновениями своей жизни. Поэтому нам по душе *Choqok* – приложение KDE 4, позволяющее подключиться к Twitter и Identi.ca, отслеживать свои события и управлять входящими и исходящими сообщениями-твитами.

Интерфейс снабжен вкладками, аналогично некоторым другим популярным клиентам в Linux, и по умолчанию настроен так, чтобы помочь вам перепрыгивать с вашей временной оси на ответы, частные сообщения и внешние микроблоги. Явной функции поиска нет, но, заглянув в меню File, вы увидите здесь возможность искать по заданным словам, пользователям или случайным тэгам.

Малость раздражает отсутствие простой возможности следовать за пользователем. Вместо этого придется нажать на имя пользователя, выбрать опцию Whois и затем нажать на кнопку Add, чтобы начать отслеживание. Недочет мелкий, но все же о нем следовало подумать. Зато вы можете легко отвечать, повторно оставлять сообщение на сервисе микроблоггинга или отмечать удачные твиты, наведя курсор на аватару пользователя и затем выбрав соответствующий значок.

Маленький шедевр

Окно поиска также позволяет переключаться с поиска по ключевым словам на поиск по твитам от нужного пользователя или адресованных ему, а также по сообщениям, содержащими имя пользователя или любые соответствующие тэги. Это более чем компенсирует лишние щелчки для следования за кем-то, и сама по себе функция поиска отличается быстротой. Один из ключевых моментов в поиске Twitter заключается в том, что он никогда не происходит «живьем», но *Choqock* обходит необходимость щелкать по Load New Posts с помощью функции автообновления, что весьма полезно, когда вы следите за особо злободневной историей.

> «Вы можете легко отвечать или отмечать удачные твиты.»



› Ищете хороший клиент Twitter для KDE? Считайте, что нашли...

Единственное, что нам еще не понравилось – интеграция с *Amarok*: она замусоривает вашу ленту тем, что вы в данный момент слушаете. Эта тактика явно разработана для того, чтобы все ваши последователи быстренько отстали от вас. Учитывая большое количество социальных сетей, встраиваемых в одно приложение, приятно видеть программу, хорошо работающую с Twitter. Мы от души рекомендуем ее, если вы используете KDE.

Программа для рисования

TuxPaint

Версия 0.9.21 **Сайт** www.tuxpaint.org

Порой нас в Башнях **LXF** радуют самые незатейливые вещи: солнышко светит, бабочки порхают, детишки резвятся… это если они не носятся вокруг с воплями. А есть ли лучший способ пресечь нездоровую активность своих отпрысков, чем попытаться приучить их к творчеству с помощью *TuxPaint*?

Он создан специально для детей, но, тем не менее, там хватает новых функций, способных заставить другие проекты позеленеть от зависти. Мы перейдем к ним через минуту, но сперва поговорим об интерфейсе. Идеология *TuxPaint* ясна с ходу: по сторонам области редактирования – ряды здоровенных кнопок, предоставляющих доступ к разным инструментам. Слева – сами инструменты, а правая сторона изменяется в зависимости от выбранного. Например, активировав Paint, вы увидите здесь кисти разной формы. Инструмент Stamp, наверное, самый впечатляющий: в него включено множество вариантов дизайна, от клякс в виде лягушек и жаб до пожарных или вороньего пугала. Здесь ваш неуемный малыш гарантированно найдет, чем заняться.

Есть где попрыгать

Можно легко создавать дополнительные линии и формы, используя кисти, которые здесь тоже предусмотрены – наш подопытный, маленький Джон, пришел в восторг от прыжков по экрану кисти Squirrel, когда мы рисовали линию с ее помощью. Оборотная сторона медали – *TuxPaint* не слишком богат шрифтами; однако детей это волнует куда меньше, чем нас. Весьма увлекательна кнопка Magic, она открывает кучу эффектов, которыми дети могут воспользоваться при создании своих шедевров; среди них – весьма полезная кисточка Grass, идеальная для домикового периода любого юного художника.

> «Кисть Squirrel прыгает по экрану, когда вы рисуете.»



› Обаятельный *TuxPaint* поможет раскрыться юному дарованию.

А если серьезно, то *TuxPaint* – отличная программа, которая введет ваших детей в мир компьютерной графики и позволит им заодно еще и развлечься. Ну, а если вы сами вдруг впадете в детство, вы теперь знаете, куда податься.

Клиент Subversion

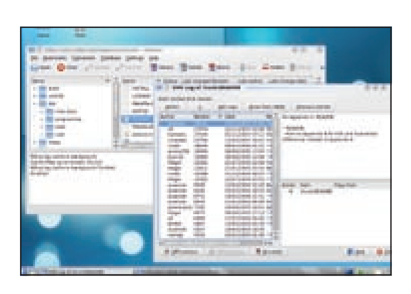
KDESvn

Версия 1.4.0 **Сайт** http://kdesvn.alwins-world.de

Эффективное управление исходным кодом – один из основных движителей разработки Open Source, предлагающий разработчикам и кодерам всего мира множество преимуществ. *CVS* отлично служил нам и до сих пор используется во множестве проектов, но многие перешли на другие системы, например, на *Git* и *Subversion*; последняя-то нас и интересует. Работа с любой из этих систем через командную строку способна оказаться тяжким бременем, и здесь вам поможет клиент, благодаря которому вы не заблудитесь в паутине связей вашего проекта.

KDESvn – клиент KDE, разработанный именно для этих целей. Его популярность доказана, и недавно вышла версия 1.4.0, что дало нам повод снова обратить на него внимание. Как полагается, улучшилась скорость, и в сам клиент была интегрирована версия 1.6 API *Subversion*, что позволяет применять новейшие функции *Subversion*. Настройка *KDESvn* – процесс несложный; если вы – пользователь Fedora, вы найдете 64-битные RPM на странице проекта, вместе с исходным кодом. У вас, понятное дело, должен быть установлен *Subversion*. Опции для *CMake* хорошо задокументированы в файле **install-cmake** – полюбопытствуйте, если ищете способы оптимизировать свой скомпилированный бинарник.

В любом случае, *KDESvn* – отличный клиент для работы и для отслеживания репозиториев *Subversion*.



› Хороший клиент для управления исходным кодом — важное условие для каждого открытого репозитория.

Менеджер альтернатив

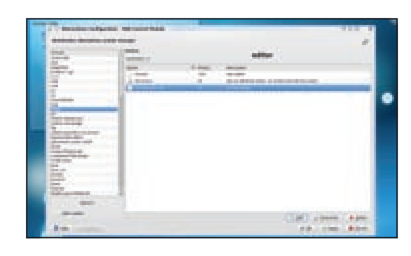
KAlternatives

Версия 0.13 **Сайт** http://freshmeat.net/projects/kalternatives

Выбор – это хорошо, но порой бывает трудно контролировать то, что навыбирал ваш дистрибутив. Вот почему стоит рассмотреть *KAlteratives*, приложение KDE, предоставляющее графическую оболочку для инструмента командной строки *update-alternatives*.

Классифицировать *KAlternatives* довольно трудно, потому что это скорее модуль расширения для *System Settings* в KDE, чем отдельное приложение; вы получаете доступ к нему через *System Settings*, а не через меню K. Это не проблема, но слегка осложняет его поиск. Перед его запуском нужно добыть права администратора, потому что он изменяет настройки по умолчанию вашей системы. Зайдя в него, вы обнаружите основной интерфейс, разработанный для того, чтобы облегчить изменение объектов по умолчанию в различных записях в директории **/usr/bin.** Например, просматривая список в Editor, вы увидите варианты *Ed*, *Nano* и *Vim*, причем *Nano* стоит по умолчанию. Прелесть *KAlternatives* в том, что можно быстро увидеть целевой исполняемый файл с описанием; при случае это поможет вам быстро внести изменения.

Запустить *KAlternatives* легко, потому что ему требуется только *kdelibs 4.1*, со встроенной поддержкой большинства популярных дистрибутивов, в том числе Debian, Ubuntu, Mandriva, OpenSUSE и Fedora. Просто выберите нужный вам во время компиляции, чтобы извлечь максимум из этого полезного инструмента.

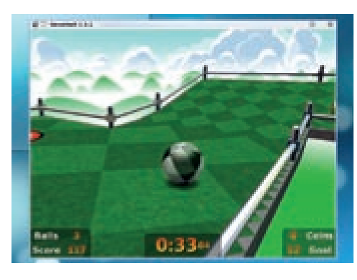


› Желаете изменить какую-то программу, ради лучшей функциональности? Тогда *KAlternatives* — для вас.

Также вышли

Новые и обновленные программы, тоже достойные внимания...

Neverball 1.5.3
Marble Madness стала еще более безумной – ведите мяч, наклоняя платформу с помощью мыши, заодно подбирая побольше монет.
http://neverball.org



› *Neverball*: наклоняйте трассу с препятствиями, катясь против часовой стрелки.

OpenVAS 2.0.5
Этот сканер уязвимостей использует многочисленные модули расширения, помогающие выявить слабые места вашей машины.
www.openvas.org

Florence 0.4.4
Экранная клавиатура для Gnome в стиле *Kvkbd*; особенно полезна для сенсорных интерфейсов.
http://florence.sourceforge.net

Frescobaldi 0.7.14
Создавайте качественные музыкальные партитуры для аранжировок с одной или несколькими партиями.
www.frescobaldi.org

Booh 0.9.2
Генерируйте статические web-альбомы, готовые к загрузке непосредственно на ваш сайт. Просто и очень быстро.
http://booh.org

Jampa 0.3.0
Музыкальный плейер для плей-листов с *aTunes* с дополнительными функциями.
http://jampa.sourceforge.net

Klick 0.12.2
То, что обязательно должны иметь все крутые музыканты: консольный метроном для проверки правильности темпа.
http://das.nasophon.de/klick

Rednotebook 0.8.6.1
Личный дневник и журнал со множеством полезных функций, чтобы не забыть о том, что вас волнует.
http://digitaldump.wordpress.com/projects/rednotebook

Начните здесь

Приготовьтесь благоговеть — это Slackware...



Читайте дальше — вы найдете всю необходимую информацию про DVD! Если вы новичок в Linux, откройте на диске файл index.html и перейдите в раздел Справка — там имеются руководства по ОС, в том числе:

» Что такое Linux?
» Что такое дистрибутив?
» Загрузка компьютера с DVD
» Разбиение жесткого диска на разделы
» Навигация по файловой системе
» Учетные записи супер- и обычного пользователя
» Работа в командной строке
» Установка программ
» Помощь онлайн

...и еще много полезного!

Майк Сондерс
Редактор диска
mike.saunders@futurenet.com

Дистрибутив Linux

Slackware 13.0

Дистрибутивы приходят и уходят, подобно временам года; но, хотя некоторым из них удается создать для себя нишу на рынке Linux, ни один не может похвастаться столь же долгой жизнью, как у Slackware. Впервые появившись в 1993 году, Slackware всегда фокусировал внимание на простоте. Пока другие крупные дистрибутивы добавляют все новые заплатки к своим пакетам и оснащают систему все новыми мастерами настройки, Slackware держится в максимальной близости к авторскому коду, и вы получаете очень чистую систему.

Этот упор на простоту означает, что у вас не будет обвешанных виджетами графических конфигураторов, появляющихся при первом намеке; программа установки работает в текстовом режиме и предполагает наличие неких начальных знаний о Linux и о разбивке жесткого диска на разделы. Если вы – новичок в мире Linux, рекомендуем обратиться к Zenwalk (переверните страницу), но если вы уже кой в чем поднаторели и жаждете новых знаний, попробуйте Slackware. Этот дистрибутив отлично подходит для того, чтобы глубже понять, как работать в Linux.

Минимальные системные требования:

» **Процессор 486**
» **64 МБ ОЗУ**
» **500 МБ места на жестком диске**

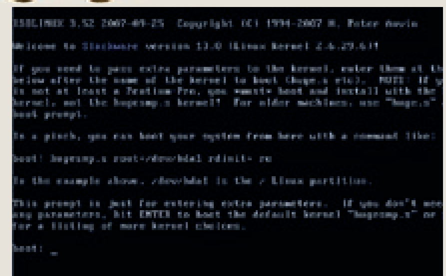
Но они идеальны только для серверных систем в текстовом режиме. Для графических настольных систем мы рекомендуем 512 МБ ОЗУ, 1-ГГц CPU и 10 ГБ места на жестком диске. Slackware поставляется с графическими окружениями KDE и *Xfce*, а также со множеством настольных приложений, инструментов разработки и серверных программ.

Важный момент по поводу разбивки диска на разделы: программа установки Slackware использует *cfdisk*, способный запугать, если вы его никогда раньше не видели. На самом деле, все очень просто: используя клавиши курсора вверх и вниз, выберите диск, который хотите разбить на разделы, затем нажимайте кнопки влево и вправо, чтобы выбрать опции внизу. Вы можете создать новые разделы и выбрать их формат – в большинстве случаев можно просто организовать один большой раздел type-83 (Linux) и раздел подкачки type-82, на 512 МБ.

Программа установки предложит вам задать пароль root. После установки вы перезагрузитесь в текстовый режим – на смежной странице вы найдете инструкции, как заставить графический рабочий стол стартовать автоматически. Конечно, постоянно работать от имени root – не самая лучшая идея, поскольку любая ошибка в команде может угробить всю систему, так что мы заодно объясним, как настроить обычные пользовательские учетные записи.



Шаг за шагом: Устанавливаем Slackware 13.0



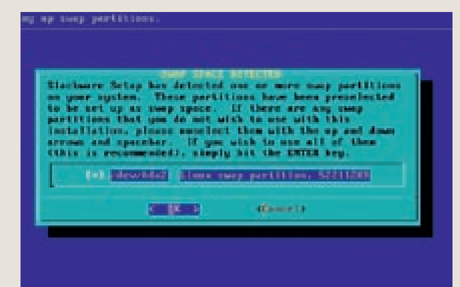
1 Загрузка

Запустите компьютер с **LXF**DVD и нажмите на Enter на начальном экране (чтобы загружаться с DVD, вам, возможно, придется изменить порядок загрузки в BIOS).



2 Создание разделов

Зайдите как суперпользователь-root (пароль не потребуется), затем введите **cfdisk**. Создайте файловую систему type-83 размером не менее 10 ГБ и раздел подкачки type-82 размером 512 МБ.



3 Программа установки

Запишите изменения и выйдите из *cfdisk*. Затем введите **setup**, чтобы запустить программу установки, выберите в меню пункт Addswap и укажите ранее созданный раздел как раздел подкачки.

Как бы мне?..

Установив Slackware с **LXF**DVD, вы, без сомнения, захотите узнать побольше о работе с этим дистрибутивом. Вот некоторые из наиболее общих задач, которые вы захотите решить, и способы их решения…

» **Запуск рабочего стола** По умолчанию Slackware загружается в текстовом режиме. Зайдите от имени root под тем паролем, который был задан в программе установки, затем введите **startx**. Вам предоставят на выбор рабочий стол либо оконный менеджер – в дальнейших инструкциях предполагается, что вы используете KDE.

» **Графика автоматом** Если вы хотите постоянно загружаться в графическом режиме, запустите текстовый редактор (в меню K > Приложения > Утилиты) и откройте файл **/etc/inittab**. В строке 24, где написано 'initdefault', измените «3» на «4». Сохраните файл и перезагрузитесь.

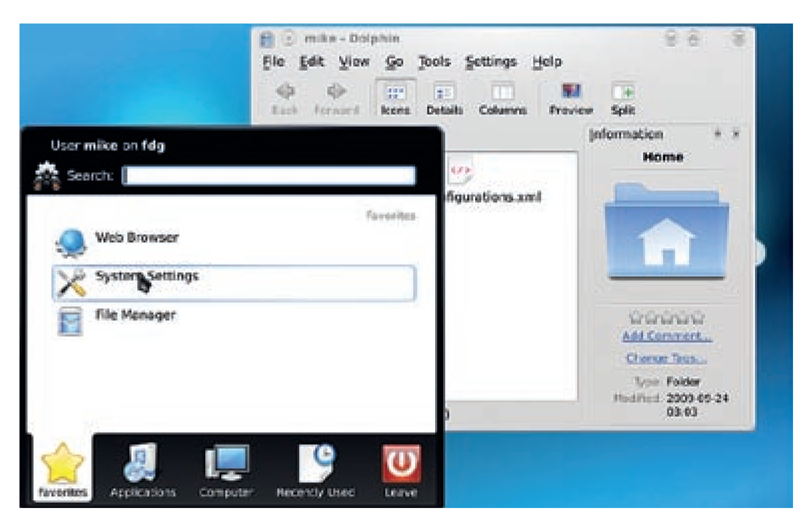
» **Добавление учетных записей обычных пользователей** Откройте терминал (Приложения > Система > Терминал) и наберите **adduser**. Введите имя пользователя и затем примите все остальные настройки по умолчанию, пока вам не предложат задать пароль. Теперь вы можете войти в систему как этот пользователь.

» **Управление пакетами** Откройте терминал, введите **su** – и вам будет предложено набрать пароль root. Наберите **pkgtool** для вызова диалоговой утилиты, которая позволит вам увидеть список пакетов, устанавливать и удалять их. Если вам нужен подход в стиле Debian, загляните в *Gslapt* и *slapt-get* в разделе **Система** на **LXF**DVD.

» **Настройка системы** Чтобы изменить интерфейс, настройки клавиатуры и мыши или сетевые настройки, нажмите на меню K > Компьютер > Параметры системы.

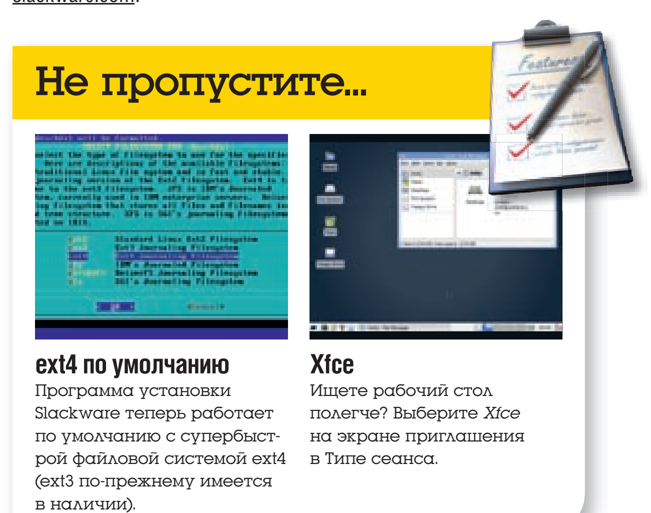
» **Выход или перезагрузка** Щелкните по меню K > Выйти и по нужной вам опции.

Версия Slackware на **LXF**DVD включает более 1000 пакетов, а если вам этого мало, www.linuxpackages.net – настоящая сокровищница того, что вы сможете установить, используя *pkgtool*, как описано выше. По причине трудоемкости сборки, Slackware не включает Gnome, однако вы можете найти отлично сделанные пакеты на www.gware.org, www.gnomeslackbuild.org и www.droplinegnome.net. Они специально собраны так, чтобы хорошо вписаться в дистрибутив.



› **Slackware включает KDE 4.2.4, но более легкие рабочие столы и менеджеры окон также есть на DVD.**

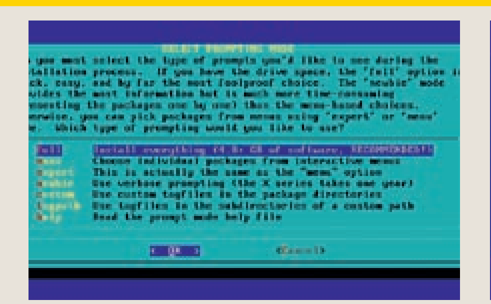
Если вам понадобится помощь по Slackware, на www.slackware.com найдутся ссылки на полезные ресурсы, или можно задать вопрос на www.linuxforum.ru. Учтите, что Slackware – по большей части детище одного человека (Патрика Фолькердинга [Patrick Volkerding]), и вы можете оказать финансовую поддержку дальнейшей разработке, купив коробочную версию на http://store.slackware.com!



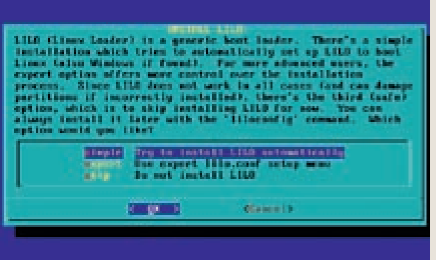
Не пропустите...

ext4 по умолчанию
Программа установки Slackware теперь работает по умолчанию с супербыстрой файловой системой ext4 (ext3 по-прежнему имеется в наличии).

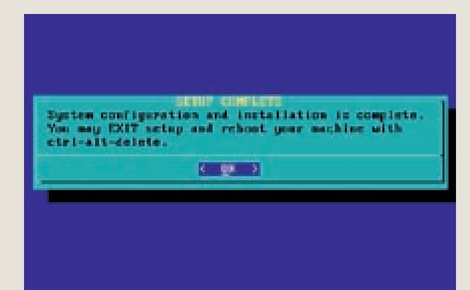
Xfce
Ищете рабочий стол полегче? Выберите *Xfce* на экране приглашения в Типе сеанса.



4 Программы
Укажите раздел диска для корневой файловой системы (/), выберите Install From DVD и отметьте нужные вам приложения. Для упрощения задачи рекомендуем опцию 'full' («полностью»).



5 Программа загрузки
Инсталлятор предложит установить *Lilo*; рекомендуем выбрать опцию «simple» и разместить *Lilo* в Master Boot Record.



6 Настройка
После ряда последних шагов настройки, например, выбора рабочего стола по умолчанию, выйдите из программы установки и нажмите Ctrl+Alt+Del для перезагрузки.

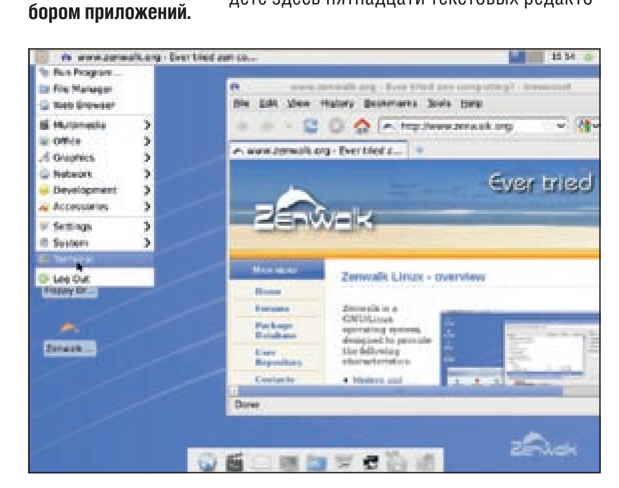
Дистрибутивы Linux

Zenwalk 6.2

Zenwalk – один из наших фаворитов среди легковесных дистрибутивов, и изначально он основывался на Slackware. Но Slackware борется за чистоту и нацелен на пользователей Linux среднего уровня, а Zenwalk большее внимание уделяет быстрой и дружественной установке для новичков.

Одной из отличительных особенностей Zenwalk является производительность, а другой – предоставление по одной программе для каждого вида задач. Вы не найдете здесь пятнадцати текстовых редакторов и семи почтовых клиентов – только комплект отобранных вручную программ в быстрой настольной среде *Xfce*. Системные требования радуют непритязательностью: работа будет ровной на Pentium III со 128 МБ ОЗУ, и вам потребуется всего лишь 2 ГБ на жестком диске. На современных системах Zenwalk просто летает.

› Zenwalk Linux объединяет отличную производительность с чистым рабочим столом *Xfce* и широким выбором приложений.



Для установки возьмите **zenwalk-6.2.iso** в разделе **Дистрибутивы** на **LXF**DVD и запишите его на CD-R. (Помните, что вы должны прожечь его как ISO-образ, а не просто скопировать на диск – если это вам непонятно, загляните в документацию своей программы для записи дисков). Потом загрузите компьютер с нового диска и следуйте инструкциям:

1 Когда появится программа установки, нажмите на Enter, чтобы выбрать раскладку клавиатуры. Для навигации применяются клавиши курсора и Enter.

2 Если вы можете отвести под Zenwalk весь жесткий диск целиком, это очень просто: выберите опцию AutoInstall [Автоустановка] – и все; переходите к шагу 4.

3 Если вы хотите установить Zenwalk вместе с другой ОС, это будет посложнее: выберите пункт Partition [Разбиение], отметьте пробелом тот диск, на который собираетесь поставить Zenwalk, и нажмите Enter. Теперь используйте клавиши курсора вверх и вниз для навигации по разделам диска, а клавиши вправо и влево – для создания и удаления разделов. Вы должны создать раздел Linux размером не менее 2 ГБ и раздел подкачки размером 512 МБ. Запишите изменения, затем выберите Install [Установить] в главном меню. Выделите созданные вами разделы и выберите установку с CD.

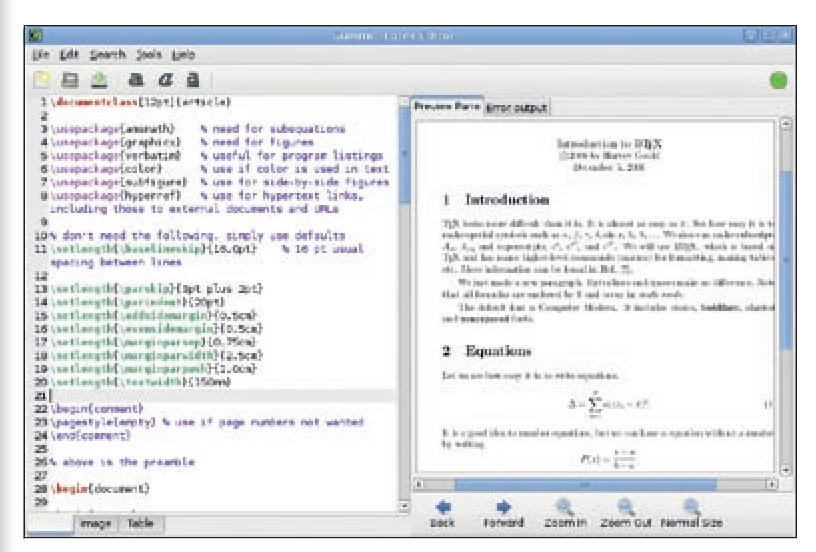
4 Начнут устанавливаться пакеты Zenwalk; это займет несколько минут. Затем вам предложат установить загрузчик – выберите опцию «simple» и установите ее в MBR.

5 В ответ на предложение перезагрузиться нажмите Ctrl+Alt+Delete. Не забудьте сперва извлечь CD с Zenwalk, иначе вы опять загрузитесь с него! Когда Zenwalk стартует, вам предложат принять лицензионное соглашение и выбрать свое местоположение.

6 Введите пароль для root (администратора). Не забудьте, что в пароле важен регистр! Затем выберите Create User [Создать пользователя] для создания учетной записи обычного пользователя – под которой вы будете работать повседневно. Введите имя пользователя, примите установки по умолчанию и задайте пароль.

Тут перед вами появится графический экран приглашения; зайдите в систему как обычный пользователь и приступайте к знакомству с Zenwalk! Внизу вы увидите панель с кнопками быстрого запуска наиболее общих программ – браузера, файлового менеджера, медиа-плейера и т.п. Значок с желтым дельфином вверху слева открывает главное меню, а зеленый значок вверху справа позволяет выйти из системы или выключить компьютер.

Вы можете настроить дистрибутив, перейдя в подменю Settings [Настройки]



› Впечатляющий результат: *Gummi* стал отличным инструментом для пользователей *LaTeX*, не успев достигнуть даже версии 0.5.





Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим:

Диски Linux Format DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция Linux Format не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

Бракованные диски

В маловероятном случае обнаружения бракованного диска Linux Format, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru для получения содействия.

и System [Система] под значком с дельфином. Более подробную информацию и помощь по работе с дистрибутивом вы найдете на www.zenwalk.org.

Операционная система

Haiku

Помните BeOS? В конце 90-х эта многообещающая ОС с ориентацией на мультимедиа считалась претендентом на солидную часть рынка настольных систем. Ее отличали быстрота, элегантность, отсутствие исторически накопленного хлама и удобство использования. Но увы: все эти плюсы не сумели перебить мощь Microsoft, и в конечном итоге BeOS канула в забвение. (Be Inc, создатели BeOS, вчинили Microsoft иск, обвиняя их в практике ведения бизнеса, направленной на подавление конкуренции, но Microsoft отделалась суммой в 23 миллиона долларов.)

Однако же BeOS настолько полюбилась многим, что спустя некоторое время появился проект-клон с открытым кодом. Изначально названная OpenBeOS, Haiku разрабатывалась много лет, и вот наконец ее команда созрела представить на суд миллионов Haiku alpha 1. Как подсказывает название, она по-прежнему находится в стадии интенсивной разработки, так что ждите ошибок и сбоев. Но это определенно отличный испытательный образец.

Haiku поставляется в виде ISO-образа, который можно записать на CD-R и загрузиться с него, но есть и более быстрый способ – использовать эмулятор типа *Qemu* (см. раздел **Система** на **LXF**DVD). Пусть *Qemu* установлен и **LXF**DVD подмонтирован в **/cdrom**:

```
cp /cdrom/Distros/Haiku/haikur1alpha1.iso.gz
gunzip haiku-r1alpha1.iso.gz
qemu -cdrom haiku-r1alpha1.iso -boot d -m 128
```

Перед вами возникнет тщательно отделанный экран загрузки Haiku, где следует выбрать 'Desktop (Live mode)'. Нажмите на значок с листком вверху справа, чтобы познакомиться со включенными программами; здесь немало инструментов рабочего стола и приложений, поставляемых по умолчанию. Есть и несколько деморoликов, подчеркивающих многозадачный характер ОС – хорошим примером служит ролик *GLTeapot*. Следите за развитием проекта на www.haiku-os.org, потому что там будет еще много интересного.

Другие программы

Настольные приложения, инструменты разработки

В разделе **Рабочий стол** вы найдете *CairoLauncher*, впечатляющее средство запуска программ, щеголяющее всевозможной анимацией и эффектами изменения масштаба. Если вы цените внешний лоск, вы обязательно должны на него взглянуть. Еще один яркий момент в этом разделе – *Gummi*, редактор *LaTeX*, написанный на Python в *GTK*. Он пока что находится посреди процесса разработки, но уже выглядит достаточно мощным – посмотрите на панель «живого» предварительного просмотра на экранном снимке слева.

А в разделе **Разработка** мы предлагаем вам новый релиз Gambas (2.18) со множеством исправленных ошибок, а также *Minimum Profit*, многогранный текстовый редактор, полностью управляемый из скриптов и поддерживающий неограниченный откат изменений [undo] наряду с подсветкой синтаксиса для широкого диапазона языков.

Если ваш браузер кажется вам чересчур медленным и громоздким, попробуйте *Midori* в разделе **Интернет**. Этот быстрый браузер на основе *GTK* основан на HTML-движке *WebKit* (подобном применяемому в *Chrome* и *Safari*) вместо *Gecko* (используемом в *Firefox*), и поддерживает расширения, написанные на C. В нем нет всех наворотов более развитых web-обозревателей, но для большинства из нас важнейшим атрибутом любого браузера является скорость, а в этом отношении *Midori* выглядит отлично.

В разделе **Интернет** имеется новый релиз *LinPhone*, программного SIP-видеофона с открытым кодом. Версия 3.2 порядком переработала интерфейс и должна лучше ладить с прокси-серверами. В раздел **Сервер** мы не могли не включить *Sshguard*, небольшую утилиту, которая следит за вашими учетными записями и блокирует IP-адреса, расцененные как опасные.



› *Midori* берет HTML-движок *WebKit* и помещает его в лаконичный интерфейс *GTK*.

› Haiku: написанная с нуля версия многообещающей мультимедийной ОС BeOS с открытым кодом, совместимая с существующими BeOS-приложениями.

Вторая сторона

Альт Линукс 5.0 Ковчег

Последние пару лет словосочетание «Альт Линукс» в основном упоминается в связи со школьным проектом, но это не значит, что разработчики известного отечественного дистрибутива не занимаются ничем другим. Напротив, все это время они трудились над «Пятой платформой» – программной концепцией, на которой будут строиться будущие релизы дистрибутивов компании, включая школьные. Длительное время «Платформа 5.0» была доступна для обывателей лишь в виде идеи, но в конце октября состоялся официальный выпуск дистрибутива Альт Линукс 5.0 Ковчег, который сделал ее осязаемой.

«Пятая платформа» зиждется на виртуализации (OpenVZ или KVM) – сторонние разработчики могут оформлять свои решения в виде контейнеров, что обеспечивает надежность системы и упрощает развертывание. Вторым столпом является технология Zeroconf – все настройки максимально автоматизированы.

Альт Линукс 5.0 доступен в виде клиентской (рабочий стол Gnome) и серверной ОС, тесно увязанных друг с другом: например, для рабочих станций обеспечивается централизованная аутентификация и интегрированное хранилище файлов на сервере. Альт Линукс 5.0 Сервер можно администрировать через web-интерфейс (Alterator), и в его возможности входит поддержка сети предприятия (включая VPN), резервирование (Bacula) и автоматическое создание массивов RAID-1.

Чтобы опробовать Альт Линукс 5.0, загрузите компьютер со второй стороны **LXF**DVD; Сервер доступен как отдельный ISO-образ в каталоге Distros. Предлагаемая 32-битная версия должна нормально работать на любом достаточно современном компьютере, но рекомендуется не менее 1 ГГц CPU/512 МБ ОЗУ/10 ГБ жесткого диска для рабочей станции и 2 ГГц CPU/1024 ГБ ОЗУ/10 ГБ диска для сервера. LXF

Информация о диске

Внимательно прочтите это перед тем, как использовать LXFDVD!

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

» **имя_программы-1.0.1.i386.rpm** – вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
» **имя_программы-1.0.1.i386.deb** – такой же пакет, но уже для Debian;
» **имя_программы-1.0.1.tar.gz** – обычно это исходный код;
» **имя_программы-1.0.1.tgz** – тот же файл, что и выше по списку: «tgz» – это сокращение от «tar.gz»;
» **имя_программы-1.0.1.tar.bz2** – тот же файл, но сжатый bzip2 вместо обычного gzip;
» **имя_программы-1.0.1.src.rpm** – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
» **имя_программы-1.0.1.i386.FC4.RPM** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
» **имя_программы-1.0.1.ppc.Suse9.rpm** – двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
» **имя_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Декабрь 2009 LXFDVD 125

LINUX FORMAT

ALT Linux 5.0 «Ковчег»

Первый отечественный дистрибутив, собранный на базе «Пятой платформы»

» Развертывание в виде виртуальных контейнеров
» Автоматическая настройка с Zeroconf
» Администрирование через web-интерфейс

Декабрь 2009 LXFDVD 125

LINUX FORMAT

Slackware 13.0

Простота, надежность, стабильность – вот те три кита, на которых держится старейший в мире дистрибутив Linux.

» **Zenwalk 6.2**
Компактный, быстрый и дружественный пользователю

» **Ингредиенты Linux-микса**
Arch и Linux From Scratch – соберите из них что-то свое

» **Haiku OS**
BeOS возвращается – теперь в альфа-версии

А ТАКЖЕ: игры и средства разработки!

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков: подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика.

DVD-приложение к журналу Linux Format № 12 (125). Декабрь 2009

Содержание

LINUX FORMAT

Сторона 1

РАБОЧИЙ СТОЛ

CairoLauncher – утилита для удобного запуска программ
Data Crow – каталогизатор фильмов, книг, изображений, программ и музыки
GCstar – приложение для управления коллекциями
Gummi – простой LaTeX-редактор на Python/GTK
Kdenlive – интуитивный и мощный видеоредактор
KMyMoney – персональный менеджер финансов
KRename – утилита для группового переименования файлов
Moll – текст-ориентированная база данных для коллекций
PeaZip – кроссплатформенный архиватор
Stuffkeeper – каталогизатор
TaskJuggler – современная система управления проектами
Tellico – это менеджер коллекций для KDE

РАЗРАБОТКА

ChucK – язык программирования звука
Gambas – среда разработки для BASIC-подобного языка
Gxmessage – утилита для создания диалоговых окон
miniAudicle – среда разработки для ChucK
Minimum Profit – редактор для программистов

ДИСТРИБУТИВЫ

Arch Linux – быстрый и гибкий дистрибутив
Haiku – открытый клон BeOS
Linux From Scratch – набор исходных кодов и инструкция по сборке
Slackware – старейший из ныне живущих дистрибутивов
Zenwalk – основанный на Slackware десктопный дистрибутив

ИГРЫ

Childsplay – комплект программ для дошкольников
FoFiX – модификация Frets on Fire
Rezerwar – головоломка с трубами
Stella – эмулятор Atari 2600
Wormux – открытый клон Worms

HOTPICKS

Areca Backup – программа для резервного копирования
Bilbo – свободный блоггинг-клиент
BMTG3 – скроллинговая одномерная стратегическая игра
Choqok – клиент микроблоггинга для среды KDE
ClamTk – интерфейс для ClamAV
CloneRNX – головоломка
Final Page – программа для работы с PDF-документами
Kalternatives – интерфейс для системы альтернатив
KDESvn – клиент для системы контроля версий Subversion
Tux Paint – программа рисования для детей

ИНТЕРНЕТ

jIRCii – кроссплатформенный IRC-клиент
Linphone – видеотелефон и клиент VoIP
LottaNZB – клиент для сетей Usenet
Midori – легкий веб-браузер на WebKit

СЕРВЕР

AgileWiki – комбинированная платформа для Web и P2P-сетей
BackupPC – высокопроизводительная система резервного копирования
Lighttpd – веб-браузер
Sshguard – инструмент для отслеживания атак на SSH

СИСТЕМА

GSlapt – GTK-версия slapt-get
Ksplice – технология для применения патчей к ядру без перезагрузки
QEMU – свободный эмулятор процессора и виртуализатор
Seahorse – приложение GNOME для управления ключами шифрования
UCK – утилита для изменения LiveCD Ubuntu
virt-manager – приложение для управления виртуальными машинами

Сторона 2

Alt Linux 5.0 Desktop
Alt Linux 5.0 Server
Десктопная и серверная версии популярного отечественного дистрибутива.

Пожалуйста, перед использованием данного диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИИ? Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕФЕКТНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь пожалуйста по адресу disks@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Тираж изготовлен ООО «Уральский электронный завод». 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПТР ВАФ 77-15.

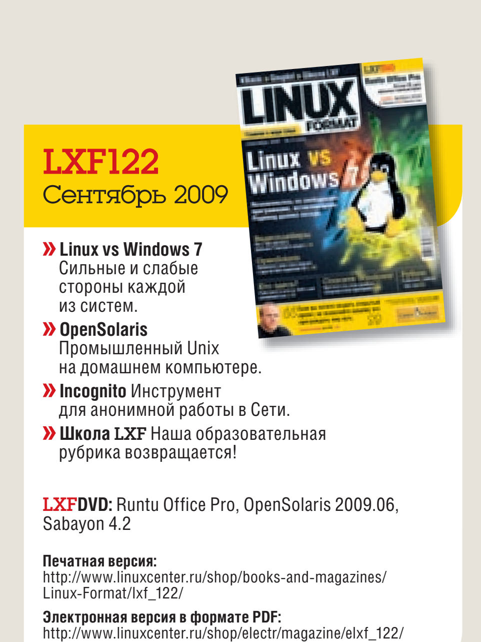




Пропустили номер?

» Мир свободного ПО богат и разнообразен, а потому далеко не все можно вместить в рамки одной статьи. Linux Format обходит эту проблему, публикуя серии статей по самым актуальным вопросам, но что делать, если вы поймали интересующий вас материал на середине? Обратитесь в Линукс-центр по адресу www.linuxcenter.ru и закажите желаемый номер журнала! Он доставляется как в печатной, так и в электронной форме, поэтому с момента открытия браузера и до получения нужного вам выпуска **LXF** может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:



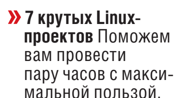
LXF122
Сентябрь 2009

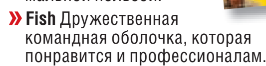
- **Linux vs Windows 7** Сильные и слабые стороны каждой из систем.
- **OpenSolaris** Промышленный Unix на домашнем компьютере.
- **Incognito** Инструмент для анонимной работы в Сети.
- **Школа LXF** Наша образовательная рубрика возвращается!

LXFDVD: Runtu Office Pro, OpenSolaris 2009.06, Sabayon 4.2

Печатная версия:
http://www.linuxcenter.ru/shop/books-and-magazines/Linux-Format/lxf_122/

Электронная версия в формате PDF:
http://www.linuxcenter.ru/shop/electr/magazine/elxf_122/





LXF123
Октябрь 2009

- **7 крутых Linux-проектов** Поможем вам провести пару часов с максимальной пользой.
- **Fish** Дружественная командная оболочка, которая понравится и профессионалам.
- **Blender** Нетривиальные техники трехмерного моделирования в свободном 3D-пакете.
- **Ekiga** Достойная альтернатива бесплатному Skype плюс видеоконференции.

LXFDVD: CrunchBang Linux 9.04, MOPS Linux 6.2.2 и коллекция свободного ПО для Windows

Печатная версия:
http://www.linuxcenter.ru/shop/books-and-magazines/Linux-Format/lxf_123/

Электронная версия в формате PDF:
http://www.linuxcenter.ru/shop/electr/magazine/elxf_123/

LXF124
Ноябрь 2009

- **Ускоряем Linux** Богатая подборка способов повысить производительность вашей ОС.
- **BSD: приручаем демона** Введение во FreeBSD для завзятых линуксоидов.
- **Drupal: сайт легко и быстро** Познакомьтесь с самой популярной открытой CMS.
- **Kdenlive** Уроки видеомонтажа свободными инструментами в свободной среде.

LXFDVD: Быстрые дистрибутивы Linux, FreeBSD 7.2 и Calculate Linux 9.9

Печатная версия:
http://www.linuxcenter.ru/shop/books-and-magazines/Linux-Format/lxf_124/

Электронная версия в формате PDF:
http://www.linuxcenter.ru/shop/electr/magazine/elxf_124/

Ну, а если вы хотите быть уверенными, что не пропустите ни один номер журнала – оформите подписку! Помните, что все подписавшиеся на печатную версию журнала через www.linuxcenter.ru получают электронную версию в подарок!

Спешите на www.linuxformat.ru/subscribe!

Диаграммы: Excel и Calc

Не секрет, что значительная часть учебников по информатике ориентирована на *Microsoft Office*. **Екатерина Пашкова** покажет, как безболезненно перейти от них к *OpenOffice.org*, на примере построения диаграмм.



Наш эксперт
Екатерина Пашкова
Преподает информатику в одной из средних школ Нижнего Новгорода, одновременно возглавляя «Педагогическое сообщество Екатерины Пашковой» (pedsovet.su).

В школьном курсе информатики разделу «Технология обработки информации в электронных таблицах» уделяется пристальное внимание: электронные таблицы изучаются в 9, 10–11 классах, а задачи по ним есть в списке задач Государственной итоговой аттестации в 9 классе и Едином государственном экзамене в 11 классе. Более того, ученикам очень нравится работать в электронных таблицах на уроках — каждый урок для них становится открытием, уроком изучения действительно *нового* материала, получения *новых* знаний.

Преподавание тем, связанных с электронными таблицами, рекомендуется вводить в старших классах (9–11 кл.), так как к этому возрасту уже все ученики могут понять и усвоить материал по обработке числовых данных: вводить формулы, использовать встроенные функции, выполнять сортировку и поиск данных, строить диаграммы. В младших же классах можно начать изучение электронных таблиц с более простых тем: эстетического оформления (форматирования данных) и создания диаграмм различного вида.

На сегодняшний день наиболее распространенным учебно-методическим комплектом (УМК) по информатике для 7–11 классов является УМК Н.Д. Угриновича. Но хотя эти учебники и включают практические задания по *OpenOffice.org*, основная часть практических работ учебника все-таки ориентирована на программные продукты компании Microsoft.

Мы решили исправить ситуацию, опубликовав статью об использовании в учебном процессе свободного офисного программного обеспечения – *OpenOffice.org Calc*, программы, предназначенной для работы с электронными таблицами.

Данный вариант электронной таблицы бывалым Excel'щикам поначалу может показаться неудобным, однако *Calc* вполне может составить конкуренцию *Excel*, а в чем-то – и превзойти его по количеству и качеству возможностей.

В школьном курсе информатики изучаются следующие основные понятия работы в электронных таблицах:

- рабочая книга, лист, столбец, строка, ячейка и т. п.;
- типы данных в ячейках;
- относительные, абсолютные и смешанные ссылки, формулы, копирование формул;
- деловая графика в электронных таблицах;
- базы данных в электронных таблицах: сортировка, поиск записей, удовлетворяющих заданным запросам.

Excel vs Calc

Исходя из этих понятий, проведем сравнение между *Excel* и *Calc*. Похожего много – принципы работы везде одинаковы: электронная таблица состоит из столбцов, именуемых латинскими буквами, и строк (нумеруемых цифрами); на их пересечении находятся ячейки (определяемые именем столбца и номером строки); диапазоны ячеек, как и в Excel, обозначаются через двоеточие (например, A1:B5 – диапазон из 10 ячеек). В самих ячейках могут находиться данные трех типов: текст, числа и формулы. Формулы традиционно начинаются со знака '=', за которым следует содержание формулы. При копировании формулы из одной ячейки в другую также изменяются ссылки на ячейки, кроме абсолютных, отмеченных знаком '$'.

> «Принципы работы в электронных таблицах одинаковы.»

Название	Excel	Calc
Сумма значений аргументов	СУММ(число1;число2,...)	SUM(число1;число2,...)
Среднее значение аргументов	СРЗНАЧ(число1;число2,...)	AVERAGE(число1;число2,...)
Возведение аргумента в степень	СТЕПЕНЬ(число;степень)	POWER(основание;степень)
Вычисление квадратного корня числа	КОРЕНЬ(число)	SQRT(число)

Отличие *Calc* от *Excel* ощущается в вопросе использования встроенных функций: меняется форма их записи. В таблице ниже приведены основные числовые функции, применяемые в рамках школьной программы.

Для вставки функции в ячейку таблицы необходимо выполнить команду Вставка > Функция в табличном процессоре. В открывшемся Мастере функций можно получить справку по использованию всех функций.

Строим диаграммы

В *Calc*, также как и в *Excel*, можно строить диаграммы и графики, предназначенные для визуального отображения числовых данных. В школьном курсе информатики изучаются следующие типы диаграмм, доступные для построения в обоих табличных процессорах:

» круговая диаграмма (предназначена главным образом для покомпонентного сравнения данных, т.е. визуального представления сведений, которые в совокупности составляют нечто целое);

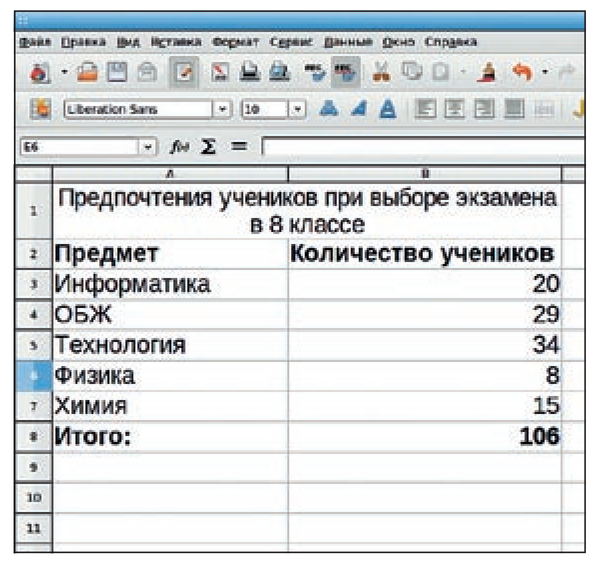
» гистограмма/линейчатая диаграмма (предназначены для временного сравнения или сравнения по позициям);

» график (предназначены для построения графиков математических функций);

» линейная диаграмма (для отображения изменения каких-то величин при переходе от одной точки к другой);

» области (для отображения изменения каждой из нескольких величин и изменения их суммы в нескольких точках);

» диаграммы с накоплением/ярусные (для наглядного сравнения суммы нескольких величин в нескольких точках при отображении вклада каждой величины в эту сумму).



Предпочтения учеников при выборе экзамена в 8 классе	
Предмет	Количество учеников
Информатика	20
ОБЖ	29
Технология	34
Физика	8
Химия	15
Итого:	106

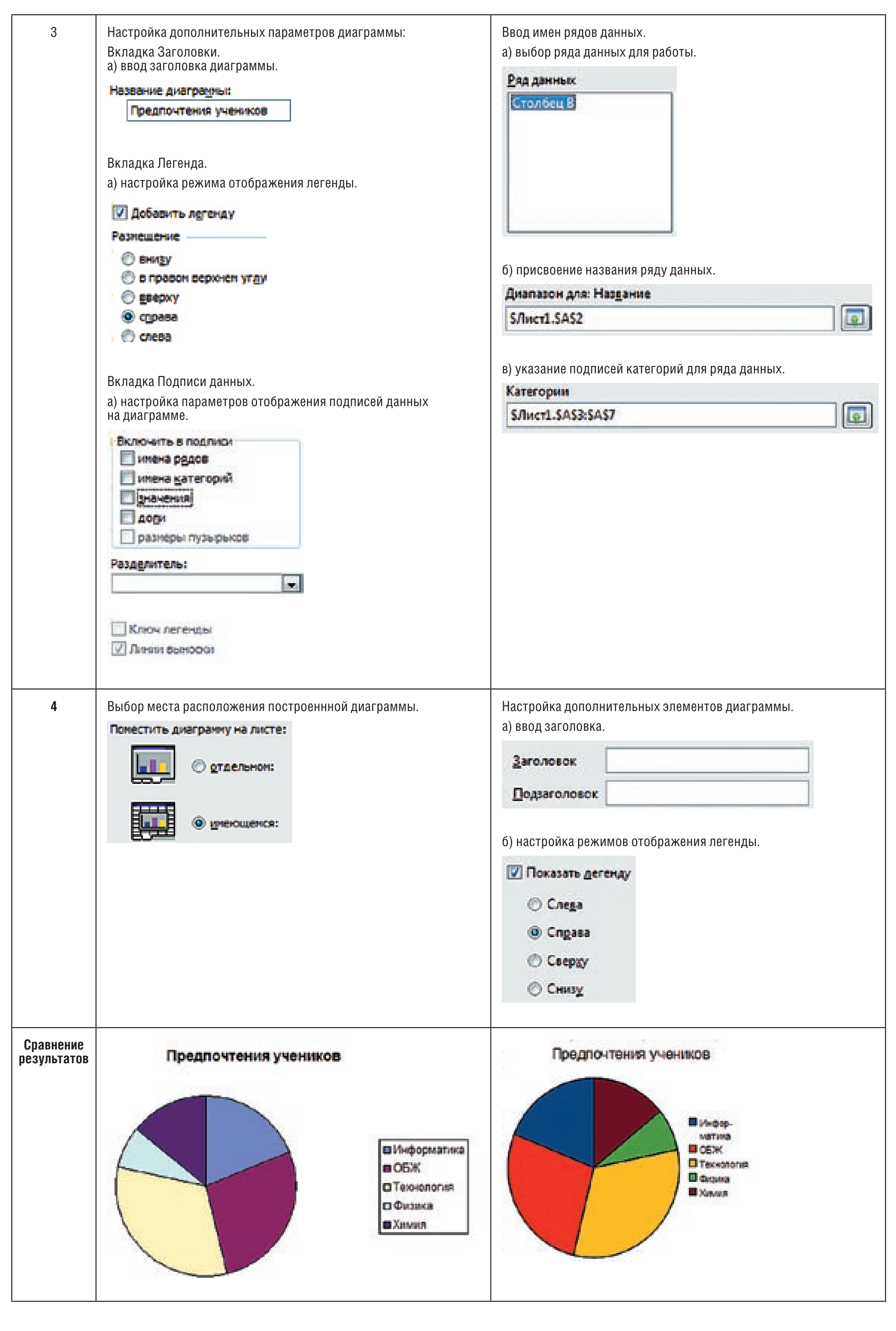
› **Рис. 1. Исходные данные для построения диаграммы в *OOo Calc*.**

Как в *Excel*, так и в *Calc* построение диаграммы осуществляется с помощью Мастера диаграммы, проводящего пользователя через ряд шагов. Открыть окно Мастера можно по команде Вставка > Диаграмма.

Рассмотрим последовательно построение круговой диаграммы в сравниваемых табличных процессорах. Для этого введем таблицу значений (см. рис. 1) и вызовем мастер указанным выше способом. Шаги, которые будет предложено пройти, сведены в таблицу.

Шаг	Excel	Calc
1	Выбор типа диаграммы: круговая.	Выбор типа диаграммы: круговая > обычная.
2	Выбор диапазона данных, по которым будет строиться диаграмма, ввод имен рядов данных: Вкладка Диапазон данных. а) указание диапазона B3:B7. Диапазон: =Лист1!$B$3:$B$7 б) указание расположения диапазона данных – в столбцах. Ряды в: строках / столбцах Вкладка Ряд. а) выбор ряда данных для работы. Ряд: Ряд1 [Добавить] [Удалить] б) присвоение имени ряду данных. Имя: =Лист1!$A$2 в) указание подписей категорий для ряда данных. Подписи категорий: =Лист1!$A$3:$A$7	Выбор диапазона данных, по которым будет строиться диаграмма: а) указание диапазона B3:B7. Диапазон данных: $Лист1.$B$3:$B$7 б) указание расположения диапазона данных – ряды данных в столбцах. Ряды данных в строках / Ряды данных в столбцах в) включение/выключения первой строки/столбца как подписи. Первая строка как подпись / Первый столбец как подпись

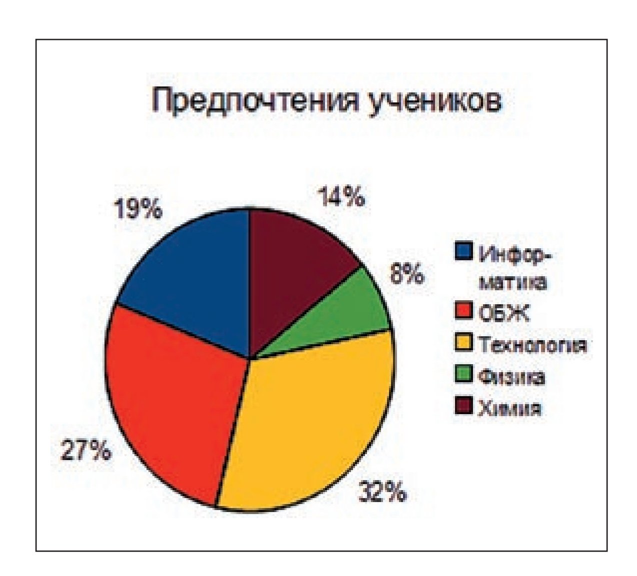
»



3	Настройка дополнительных параметров диаграммы: Вкладка Заголовки. а) ввод заголовка диаграммы. Название диаграммы: Предпочтения учеников Вкладка Легенда. а) настройка режима отображения легенды. Добавить легенду Размещение внизу в правом верхнем углу вверху справа слева Вкладка Подписи данных. а) настройка параметров отображения подписей данных на диаграмме. Включить в подписи имена рядов имена категорий значения доли размеры пузырьков Разделитель: Ключ легенды Линии выноски	Ввод имен рядов данных. а) выбор ряда данных для работы. Ряд данных Столбец B б) присвоение названия ряду данных. Диапазон для: Название $Лист1.$A$2 в) указание подписей категорий для ряда данных. Категории $Лист1.$A$3:$A$7
4	Выбор места расположения построеннной диаграммы. Поместить диаграмму на листе: отдельном: имеющемся:	Настройка дополнительных элементов диаграммы. а) ввод заголовка. Заголовок Подзаголовок б) настройка режимов отображения легенды. Показать легенду Слева Справа Сверху Снизу
Сравнение результатов	Предпочтения учеников Информатика ОБЖ Технология Физика Химия	Предпочтения учеников Информатика ОБЖ Технология Физика Химия

Как видно, алгоритмы построения диаграмм в данных табличных процессорах практически не отличаются. Небольшая разница в том, что шаги 3 и 4 не соответствуют друг другу, а также в способах задания подписей данных: в *Excel* применить различные варианты отображения подписей данных можно уже на шаге 3, а в *Calc* – только в режиме правки построенной диаграммы.

Для включения режима правки диаграммы в *Calc* необходимо выполнить команду Правка контекстного меню диаграммы; в *Excel* правку диаграммы можно производить после построения.



› **Рис. 2. Круговая диаграмма с надписями в виде процентов.**



› **Рис. 4. Парабола, построенная по заданному нами набору точек.**

Для выполнения различных модификаций диаграмм в табличных процессорах нужно вызвать контекстное меню диаграммы и выбрать в нем необходимые пункты. Например, чтобы добавить в *Calc* надписи данных, необходимо выбрать в контекстном меню пункт Свойства объекта (в режиме правки диаграммы) и указать соответствующие параметры отображения надписей данных на вкладке Надписи данных. К примеру, для отображения надписей данных в виде процентов выбирается пункт «Показать значение как процент». Диаграмма примет вид, показанный на рис. 2.

Приведенным выше способом строятся и диаграммы других типов: гистограммы, линейчатые, области, диаграммы с накоплением и т.д. Отличий в построении диаграммы по нескольким рядам данных в *Excel* и *Calc* нет.

Графики — это просто!

Рассмотрим построение диаграмм типа «график». В *Excel* для построения графиков математических функций существует специальный тип диаграмм, в *Calc* для этих целей мы будем пользоваться типом диаграмм Линии > Только линии, включая при этом режим Сглаживания линий.

Построим график параболы **$y=x^2$**. Для этого необходимо задать область построения графика в табличной форме (рис. 3).

В качестве диапазона данных, по которым мы будем производить построение графика, выберем диапазон значений **y**, то есть B2:H9; в качестве Категорий ряда данных выберем значения **x**; зададим дополнительные данные в Мастере диаграмм и получим полноценный график математической функции (рис. 4).

С помощью графиков функций можно также решать системы уравнений. К примеру, решим систему уравнений

$$\begin{cases} y = x^3 \\ y = 2x + 1 \end{cases}$$

Построим таблицу значений функции **y** и на ее основе создадим диаграмму типа Линии > Только линии, выбрав диапазон данных B2:H3 – в строках, а в качестве ряда данных – диапазон B1:H1 (см. рис. 5).

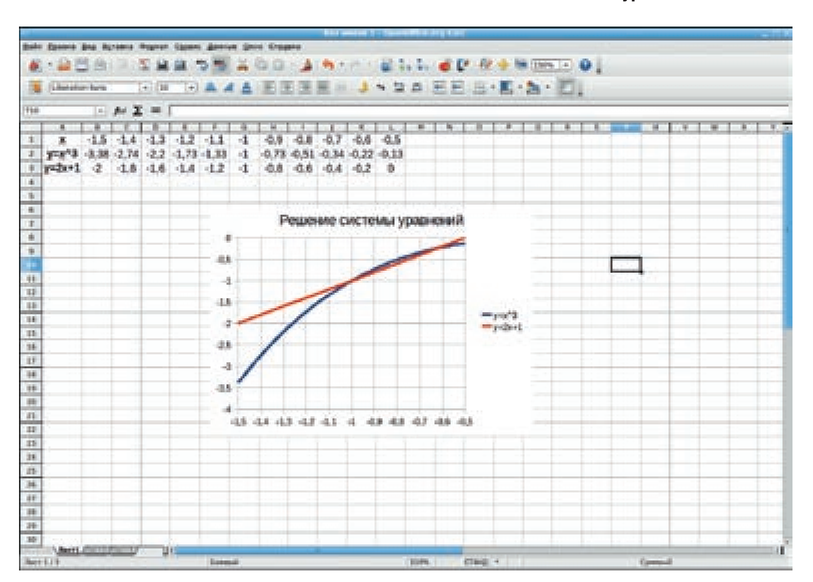
	A	B	C	D	E	F	G	H
1	x	-3	-2	-1	0	1	2	3
2	y	9	4	1	0	1	4	9

› **Рис. 3. Набор точек, по которым будет строиться парабола.**

Решение системы уравнений находится на пересечении построенных графиков, то есть в точке (-1,-1).

Подведем итоги

Итак, в данной статье мы рассмотрели основные задачи школьного курса информатики, решаемые с помощью электронных таблиц, и провели сравнение между использованием *Excel* и *Calc* для решения этих задач. Работа в программах строится на одних и тех же принципах: оба приложения отвечают требованиям, предъявляемым к табличным процессорам, поэтому переход от использования от одной программы к другой является безболезненным и не требующим больших затрат времени на изучение индивидуальных особенностей. LXF



› **Рис. 5. Исходные данные для решения системы уравнений.**

Таблицы как база данных

Отличий при работе с электронными таблицами как с базами данных в *Excel* и *Calc* почти нет. В рамках школьного курса информатики ученики должны научиться выполнять поиск данных по определенным запросам к базе данных с помощью фильтров. Фильтрация списка с помощью Автофильтра как в *Excel*, так и в *Calc* возможна через меню Данные > Фильтр > Автофильтр при задании требуемых параметров фильтрации. Работа в программах строится аналогично: можно выбрать требуемый фильтр из предложенного списка или же построить его самому в расширенном режиме.



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ
Главный редактор
Валентин Синицын info@linuxformat.ru
Литературный редактор
Елена Толстякова
Переводчики
Илья Аввакумов, Александр Бикмеев, Юлия Дронова, Светлана Кривошеина, Александр Казанцев, Алексей Опарин, Валентин Развозжаев, Татьяна Цыганова
Редактор диска
Александр Кузьменков
Верстка, допечатная подготовка
Сергей Рогожников
Технический директор
Денис Филиппов
Генеральный директор
Павел Фролов
Учредители
Частные лица
Издатели
Виктор Федосеев, Павел Фролов

Отпечатано в типографии «Взлет»
196644, Санкт-Петербург, Колпинский район, пос. Саперный, тел. (812) 462-85-85
Заказ

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ
Редактор Пол Хадсон [Paul Hudson] paul.hudson@futurenet.com
Редактор обзоров Грэм Моррисон [Graham Morrison] graham.morrison@futurenet.com
Редактор диска Майк Сондерс [Mike Saunders] mike.saunders@futurenet.com
Художественный редактор Эфраин Эрнандес-Мендоса [Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com
Литературный редактор Эндрю Грегори [Andrew Gregory] agregory@futurenet.com
Штатный автор Боб Мосс [Bob Moss] bob.moss@futurenet.com
Подготовка материалов
Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Энди Ченнел [Andy Channelle], Марко Фиоретти [Marco Fioretti], Майкл Дж. Хэммел [Michael J. Hammel], Энди Хадсон [Andy Hudson], Джульетта Кемп [Juliet Kemp], Стив Прайс [Steve Price], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Ник Вейч [Nick Veitch], Евгений Балдин, Андрей Боровский, Алексей Джежора, Олег Дмитриев, Евгений Крестников, Дмитрий Мусаев, Екатерина Пашкова, Ольга Попова, Андрей Прахов, Алексей Федорчук
Художественные ассистенты: Ник Эспелл [Nick Aspell], Фил Хейкрафт [Phil Haycraft], Салли Хендерсон [Sally Henderson], Сара Николь [Sarah Nicol]
Иллюстрации: Крис Хедли [Chris Hedley], Крис Винн [Chris Winn]

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel +44 01225 442244 Email: linuxformat@futurenet.com
РОССИЯ:
Санкт-Петербург (редакция):
Лиговский пр., 50, корп. 15
Тел.: +7 (812) 309-06-86
Представительство в Москве:
Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)
Тел/факс: +7 (499) 271-49-54
По вопросам сотрудничества, партнерства, оптовых закупок:
partner@linuxcenter.ru



BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO



В январском номере

Linux — это красиво

Надоел коричневый? Пройдитесь по своему рабочему столу руками стилиста – и пусть Windows 7 с Mac OS X будут посрамлены!

5 ценных советов

Полагаете, что вы уже знаете в Linux все? Мы найдем, чем удивить вас – как насчет абсолютно тихого системного блока или практического использования NILFS?

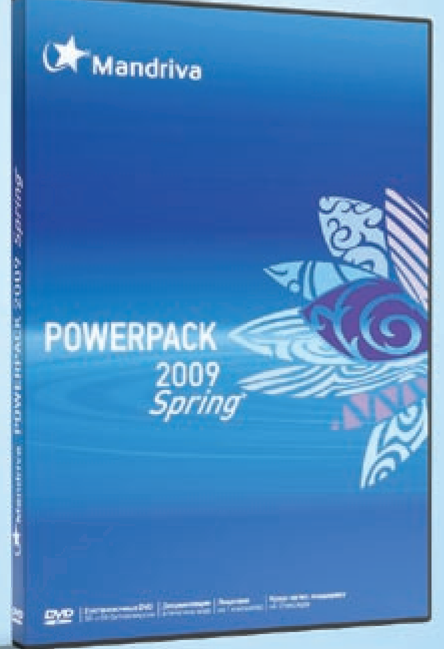
Сетевая безопасность

Защищена ли ваша сеть? В безопасности ли она? На эти и многие другие вопросы вам поможет ответить *Wireshark*.

Содержание последующих выпусков может меняться без уведомления.



Mandriva Powerpack 2009.1 Spring



Дистрибутив Mandriva Powerpack 2009.1 включает в себя набор офисных и серверных приложений, и подходит для установки на офисной или домашней рабочей станции и на сервере. Mandriva Powerpack 2009.1 является отличным вариантом для миграции на GNU/Linux новых пользователей, и в то же время удовлетворяет запросы опытных пользователей и администраторов.

Совместимость с 1С

Входящий в дистрибутив wine@etersoft local дает возможность работы с популярными отечественными бизнес-приложениями (1С, «Гарант», «Консультант» и т. д.), также дистрибутив совместим с серверными версиями 1С для GNU/Linux.

Офисные приложения

В дистрибутив входит OpenOffice.org, интернет-приложения, графические, мультимедийные приложения, ПО для верстки и другие офисные приложения.

Служба каталогов

Кроме традиционных серверных приложений, в Mandriva Powerpack входит продукт Mandriva Directory Server (аналог контроллера домена Windows NT4) — простой в использовании инструмент для ведения каталога пользователей и прав доступа к общим ресурсам локальной сети предприятия и управления сетевыми сервисами (сервера DNS, DHCP, SAMBA, Proxy, Mail и др.).

Корпоративные продукты

Mandriva Directory Server

Mandriva Directory Server (MDS) — это простой в использовании инструмент для централизованного управления учетными записями и конфигурацией таких служб, как электронная почта, прокси сервер, служба доменных имен. MDS является основой для системы управления идентификацией пользователей, разграничения доступа пользователей к интернет ресурсам, квотирования почтовых ящиков, полностью заменяет Microsoft Windows NT4, IIS, m-Daemon, работает с Windows, Linux и Mac.

Linbox Rescue Server

Linbox Rescue Server (LRS) — пакет программ, предоставляющий функции локального и удаленного управления ИТ-инфраструктурой предприятия. LRS включает функции инвентаризации программного и аппаратного обеспечения, удаленного управления компьютерами и серверами, а также резервного копирования. Администратор может управлять ИТ-инфраструктурой через веб-браузер с любого локального или удаленного компьютера.

Сертифицировано ФСТЭК

Mandriva Powerpack 2008 Spring

Дистрибутив предназначен для рабочей станции или небольшого сервера, включает необходимые офисные, графические, мультимедийные, интернет-приложения и серверное ПО.

Mandriva Corporate Server 4 Update 3

Дистрибутив Mandriva Corporate Server предназначен для создания корпоративного сервера, на базе продукта можно создать интернет-сервер, веб-сервер, почтовый, сервер печати, сервер баз данных, сервер приложений и др.

Mandriva Flash

Mandriva Flash — дистрибутив GNU/Linux, загружающийся и работающий прямо с USB-носителя. Дистрибутив включает необходимый набор офисных приложений и допускает обновление и установку новых приложений, представляя собой полноценное мобильное рабочее место для работы на любом компьютере, поддерживающем загрузку с USB-устройства.

Наименование	Код Linuxcenter.Ru	Цена, руб.
Для рабочей станции		
Mandriva Powerpack 2009.1 Spring (DVD-box, техническая поддержка 12 месяцев)	lc4942	1900
Mandriva Powerpack 2009.1 Spring (DVD-box, техническая поддержка 6 месяцев)	lc4923	1300
Мобильное рабочее место		
Mandriva Flash 2008.1 (Mandriva Linux на 8 Гб USB накопителе)	lc3215	1875
Корпоративные продукты		
LRS Сервер (Linbox Rescue Server, платформы Linux/Windows) (лицензии на рабочие станции приобретаются отдельно)	lc2856	1690
Техническая поддержка на Mandriva Directory Server на 1 год	lc3560	5550
Лицензия и техническая поддержка на Mandriva Powerpack 2009.1 Spring на 50 компьютеров	lc5105	18 750
Пакет начальной поддержки Linbox Rescue Server	lc2821	99 000
Сертифицированные ФСТЭК продукты		
Сертифицированный ФСТЭК Mandriva Powerpack 2008 Spring	lc3408	2900
Сертифицированный ФСТЭК Mandriva Flash 2008.1 (на 4GB USB накопителе)	lc3409	4900
Сертифицированный ФСТЭК Mandriva Corporate Server 4.0 Update 3	lc3410	10 050

MANDRIVA LINUX ЯВЛЯЕТСЯ ОДНИМ ИЗ ПОПУЛЯРНЕЙШИХ МИРОВЫХ ДИСТРИБУТИВОВ ОПЕРАЦИОННОЙ СИСТЕМЫ GNU/LINUX

Главные преимущества дистрибутивов Mandriva — дружелюбный интерфейс, простота настройки, возможность легкого перехода пользователей с Windows на GNU/Linux, совместимость с широким спектром аппаратного обеспечения.

Дистрибутивы Mandriva Linux сертифицированы по требованиям ФСТЭК по 5 классу для СВТ и 4 уровню контроля НДВ, что дает возможность использовать их при работе с конфиденциальной информацией и персональными данными граждан.

Множество российских государственных и коммерческих организаций успешно используют Mandriva Linux на серверах и рабочих станциях. Среди них: Правительство Московской области, администрация Черниговского района Приморского края, Министерство финансов республики Саха (Якутия), группа компаний ИМАГ и многие другие.

С вопросами по продуктам Mandriva вы можете обращаться в Mandriva.Ru
Тел.: (812) 309-06-86, (499) 271-49-55
Электронная почта: info@mandriva.ru
www.mandriva.ru

Также в Mandriva.Ru доступны:
- литература по Mandriva Linux;
- услуги по установке, настройке и технической поддержке корпоративных продуктов Mandriva.